Extending CAS with Algebraic Reductions

Author: Zongzhe Yuan

Supervised by: Timothy G. Griffin

UNIVERSITY OF CAMBRIDGE

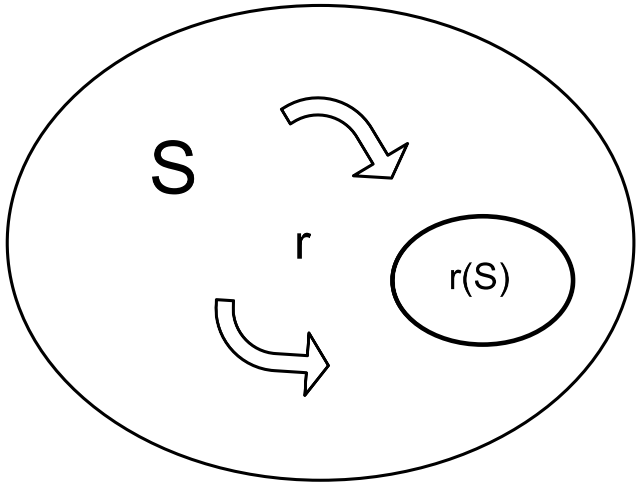
Outline

- Introduction to Reduction
- Introduction to Routing Problem
- Main Contributions
  - Full Reduction Representation
  - Generalized Reduction
  - Predicate Reduction
  - Example Applications
  - Coq Implementation

UNIVERSITY OF
CAMBRIDGE

Reduction (Classical)

By Wongseelashote in 1979



- Function **r** on the problem set **S**
- Congruence $\forall a, b \in S, a =_S b \rightarrow r(a) =_S r(b)$
- Idempotent $\forall a \in S, r(a) =_S r(r(a))$
- Left Invariant $\forall a, b \in S, r(a) \oplus b =_S a \oplus b$
- Right Invariant $\forall a, b \in S, a \oplus r(b) =_S a \oplus b$

UNIVERSITY OF CAMBRIDGE

Traditional Representation

- Representation

$$\{x \in S | r(x) =_S x\}$$

$$\forall a, b \in S, a \oplus_r b =_S r(a \oplus b)$$

- Example $min_{\leq} \equiv \{x \in S | \forall y \in S, \neg y \leq x\}$

- Hard to be implemented

$$r_2(r_1(S)) \equiv \{y \in \{x \in S | r_1(x) = x\} | r_2(y) = y\}$$

- Motivate us to create the full reduction representation

Full Reduction Representation

- Traditional Representation on Semiring

$$(S, =_S, \oplus) \longrightarrow (\{x \in S | r(x) =_S x\}, =_S, \oplus_r)$$

- Full Reduction Representation on Semiring

$$(S, =_S, \oplus) \longrightarrow (S, =_S^r, \oplus^r)$$

- Full Reduction Representation

$$\forall a, b \in S, a =_S^r b \equiv r(a) =_S r(b)$$

- $\forall a, b \in S, a \oplus^r b \equiv r(r(a) \oplus r(b))$

Isomorphism

- Isomorphic on the equality for reflexive, symmetric, congruence and transitive properties

- Isomorphic on the binary operator for associative, commutative, selective, congruence and distributive.

- Proofs only need the idempotent and congruence properties for the reduction — Detail proof can be found in Coq file

- Conclusion: two representations represent the same problem

Internet Routing Problem

- Problem for RIP like algorithm: can't start from arbitrary states
- Use BGP like algorithm, Adding Explicit Path to Shortest Path

$$spwp \equiv AddZero(\infty, (\mathbb{N}, min, +) \overrightarrow{\times} path(E))$$

- Adding reduction to eliminate problems

$$spwp \equiv red_{r_2}(AddZero(\infty, (\mathbb{N}, min, +) \overrightarrow{\times} epath(E)))$$

UNIVERSITY OF CAMBRIDGE

Classical Reduction

- Reduction with the properties of Congruence, Idempotent, Left/right invariant.

- Elementary Path reduction on **min** operator is not classical, no left/right invariant properties.

$$p_1 = \{a, a\}, p_2 = \{a, b, c\}$$

$$min(p_1, p_2) = p_1, min(r(p_1), p_2) = p_2$$

- Motivate us to generalize the definition of reduction.

UNIVERSITY OF
CAMBRIDGE

Generalized Reduction

- Get rid of left/right invariant properties
- Only have influence on associative and distributive properties — but only a sufficient condition
- Derive pseudo-associative and pseudo-distributive properties — isomorphic to associative and distributive

$$\forall a, b, c \in S, r(r(r(a) \oplus r(b)) \oplus r(c)) = r(r(a) \oplus r(r(b) \oplus r(c)))$$

$$\forall a, b, c \in S, r(r(r(a) \oplus r(r(r(b) \oplus r(c))))) = r(r(r(r(r(a) \oplus r(b))) \oplus r(c)))$$

UNIVERSITY OF CAMBRIDGE

Predicate Reduction

- A kind of Generalized Reduction
- The reduction is defined on a predicate

$$r_p(a) \equiv \begin{cases} c & P(a) \\ a & otherwise \end{cases}$$

- Decompositional — Associative and Distributive

$$\forall a, b \in S, P(a \oplus b) \rightarrow P(a) \bigvee P(b)$$

- Compositional — Classical

$$\forall a, b \in S, P(a) \bigvee P(b) \rightarrow P(a \oplus b)$$

- Preserve Order — Classical on Identity

$$\forall a, b \in S, a \leq b \wedge P(a) \rightarrow P(b)$$

Example Applications

- Min Plus With Ceiling $\forall n \in \mathbb{N}, P(n) \equiv n \geq ceiling$
- Elementary Path $\forall p \in Path, P(p) \equiv dup(p)$
- Reduce Annihilator

$$\forall (n, p) \in \mathbb{N} \times (c + Path), P((n, p)) \equiv n = ceiling \vee p = inl(c)$$

- Final Path Problem Construction

$$(\mathbb{N} \times (c + Path), (min^r \bar{\times} min^r_d)_a, (+^r \times concat^r_d)_a, (ceiling, inl(c)), (0, inr([])))$$

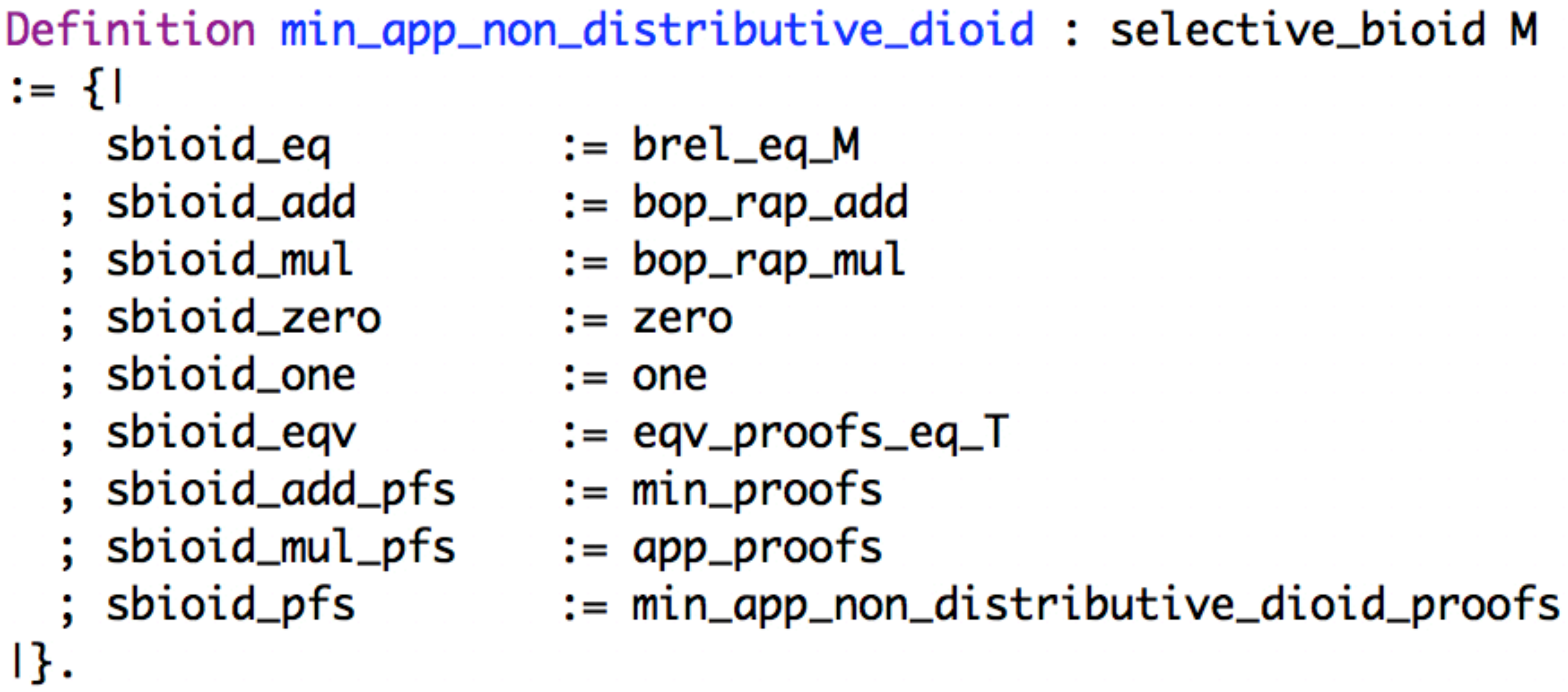
UNIVERSITY OF CAMBRIDGE

Coq Implementation

- Reason the Properties of Classical Reduction
- Prove the Isomorphism Between Full Reduction and Traditional Representation
- Prove the Properties of Generalized Reduction
- Prove the Properties of Predicate Reduction
- Construct Reduction Instance and Construct Path Problem Semiring

UNIVERSITY OF CAMBRIDGE

Coq Code Examples



```
Definition min_app_non_distributive_dioid : selective_bioid M
:= {|
    sbioid_eq         := brel_eq_M
  ; sbioid_add        := bop_rap_add
  ; sbioid_mul        := bop_rap_mul
  ; sbioid_zero       := zero
  ; sbioid_one        := one
  ; sbioid_eqv        := eqv_proofs_eq_T
  ; sbioid_add_pfs    := min_proofs
  ; sbioid_mul_pfs    := app_proofs
  ; sbioid_pfs        := min_app_non_distributive_dioid_proofs
|}.
```

Summary

Questions?