Q*cert

A platform for specifying and verifying query compilers

Challenges?
Precise Language Semantics
Long Compilation Pipeline
Query Optimizer

What for?
Correctness guarantees
New Languages (e.g., DSLs)
Education

How?
Formal Specification
Mechanized Proof
Code Extraction

Algebraic Equivalence

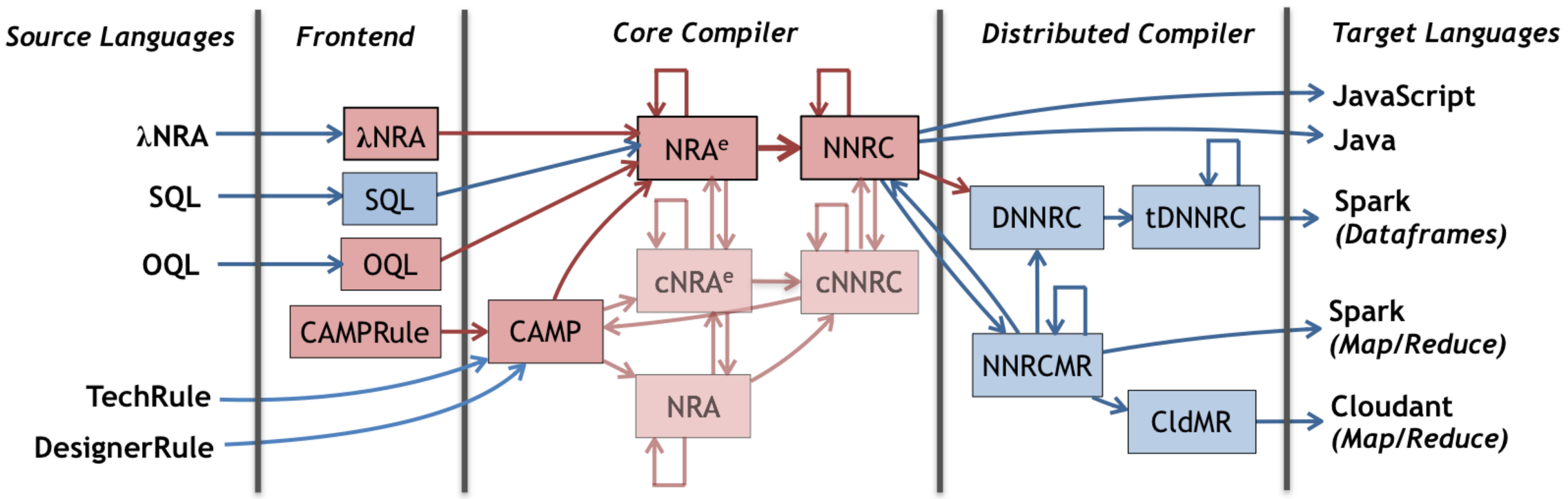
```
(* Selection distributes over union *)
Lemma select_union_distr q₀ q₁ q₂ :
    σ⟨ q₀ ⟩(q₁ ∪ q₂) ≡ σ⟨ q₀ ⟩(q₁) ∪ σ⟨ q₀ ⟩(q₂).
Proof.
  … (* proof omitted *)
Qed.
```

Functional Rewrite

```
(* Selection over union push-down *)
Definition select_union_distr_fun q :=
  match q with
  | NRAEnvSelect q0 (NRAEnvBinop AUnion q1 q2) =>
      NRAEnvBinop Aunion
        (NRAEnvSelect q0 q1) (NRAEnvSelect q0 q2)
  | _ => q
end.
```

Correctness Proof

```
(* Selection over union push-down is correct *)
Lemma select_union_distr_fun_correctness q₀ q₁ q₂ :
 select_union_distr_fun q ≡ q.
Proof.
 Hint Rewrite select_union_distr : envmap_eqs.
 prove_correctness q.
Qed.
```



SQL: Structured Query Language
OQL: Object Query Language
λNRA: NRA with Lambdas

CAMPRule: Rule Macros for CAMP
TechRule: ODM technical rules
DesignerRule: ODM designer rules

NRA: Nested Relational Algebra
NRA^e: NRA with Environments
cNRA^e: Core NRA^e
NNRC: Named Nested Relational Calculus
cNNRC: Core NNRC
CAMP: Calculus of Aggregating Matching Patterns

DNNRC: Distributed NNRC
tDNNRC: Typed DNNRC
NNRCMR: NNRC + Map/Reduce
CldMR: NNRC + Cloudant Map/Reduce

The Coq Proof Assistant

https://github.com/querycert