

Q*cert

A platform for specifying and verifying query compilers

J. Auerbach, M. Hirzel, L. Mandel, A. Shinnar, J. Siméon
IBM Research

Challenges?

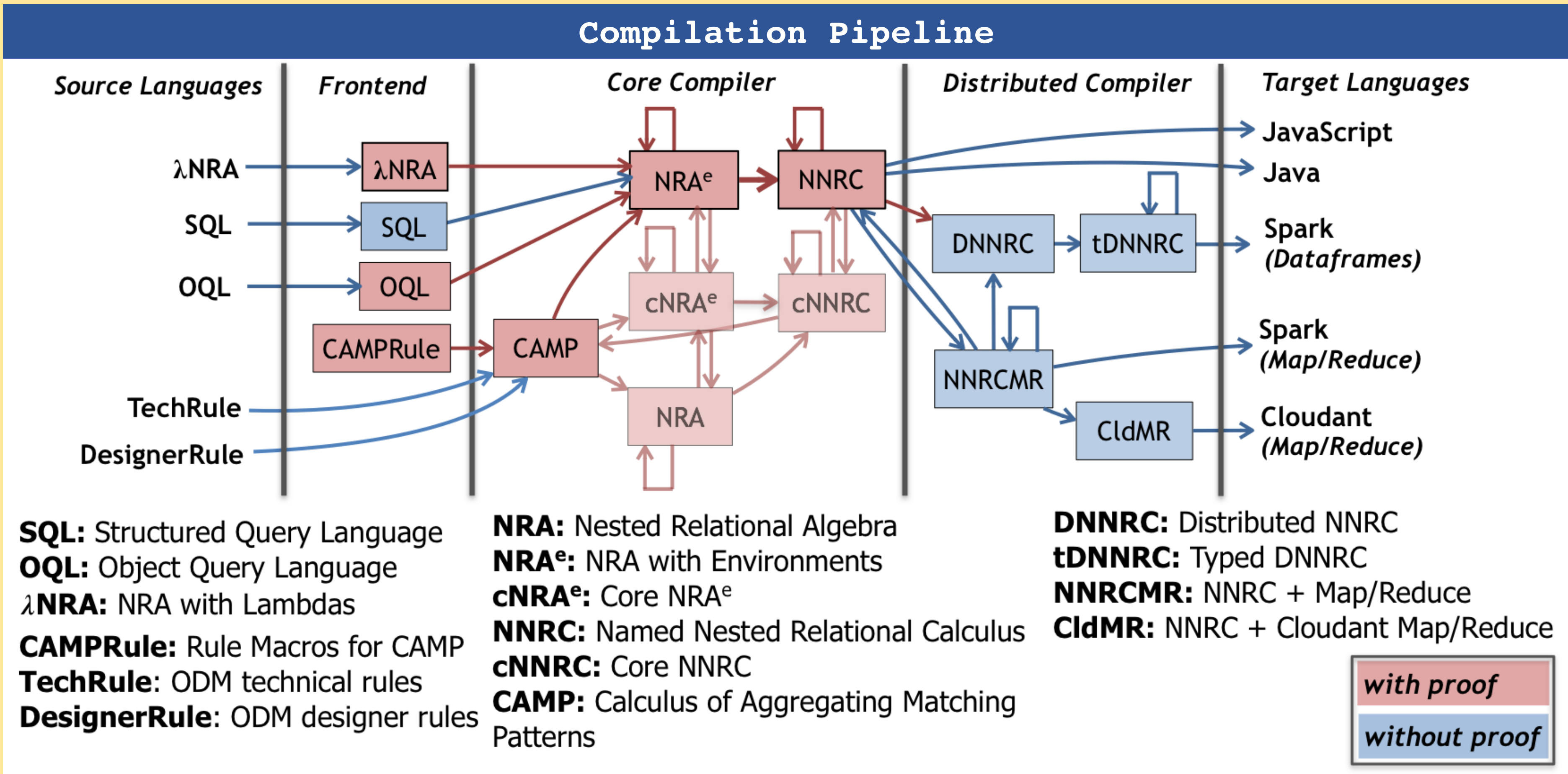
Precise Language Semantics
Long Compilation Pipeline
Query Optimizer

What for?

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

How?

Formal Specification
Mechanized Proof
Code Extraction



Algebraic Equivalence

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

Functional Rewrite

```
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

```
Property 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.
```

Features

Nested Data Model with Objects
Type Checking
Aggregate Queries (includes TPC-H)
Configurable Optimizer
External Types and Functions
JSON Support

Compiler Extraction

