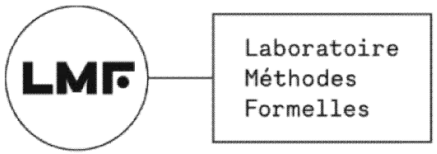
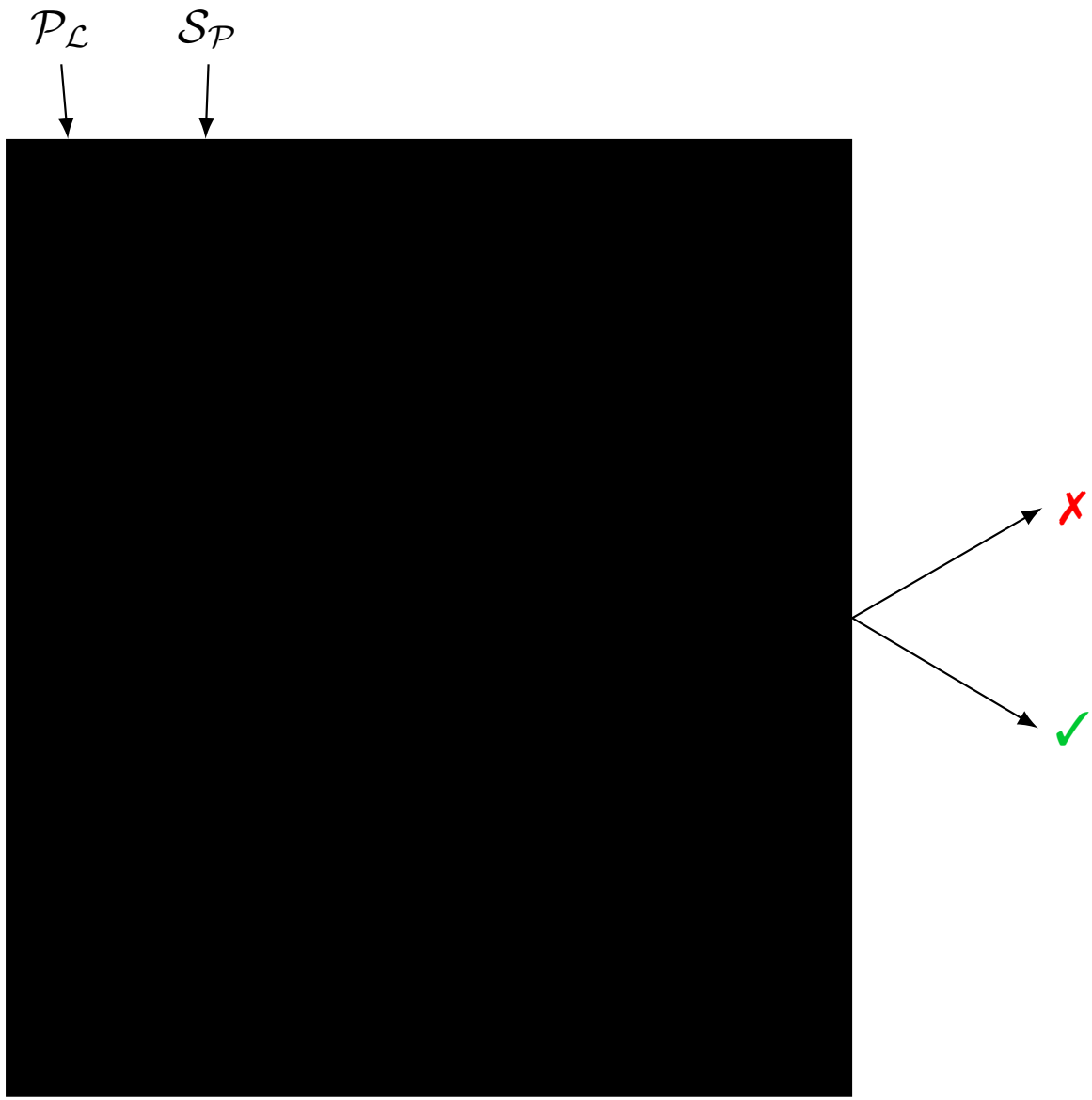
A language-independent ecosystem for program verification

Amélie LEDEIN

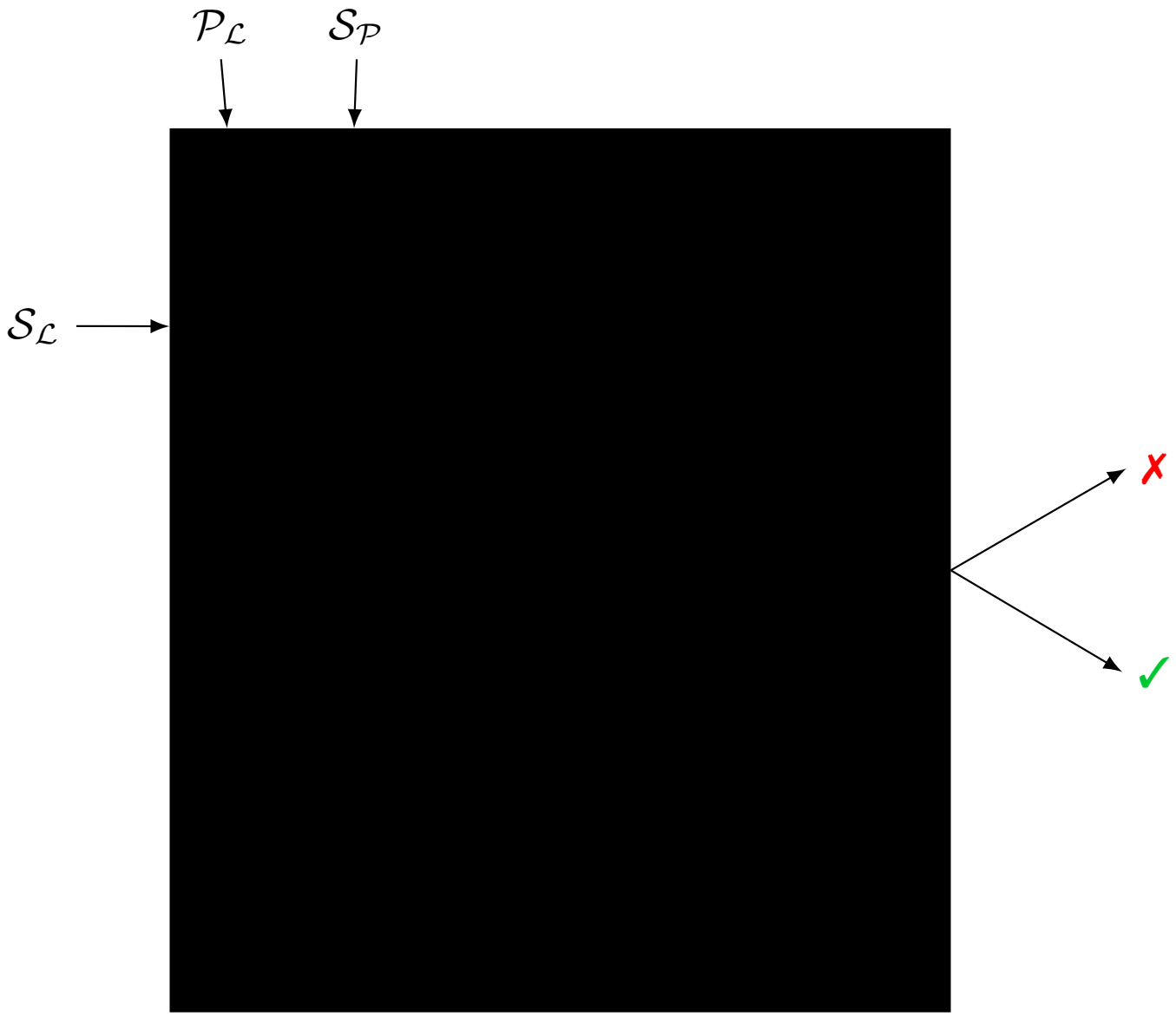
supervised by Catherine DUBOIS and Valentin BLOT



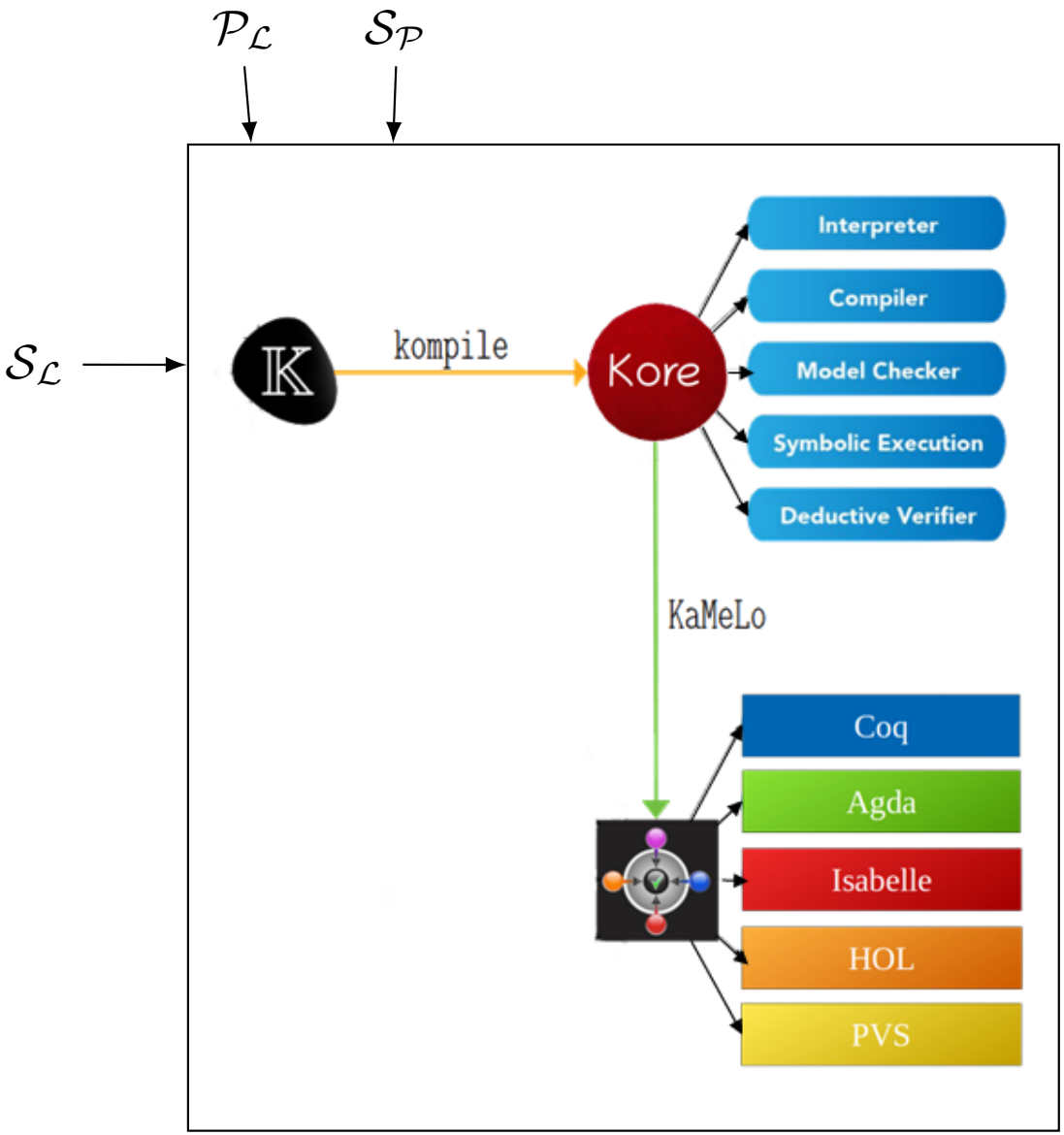
A way to do program verification



A way to do program verification



A way to do program verification



How to do that?

- **Objectives:**

- **Milestones:**

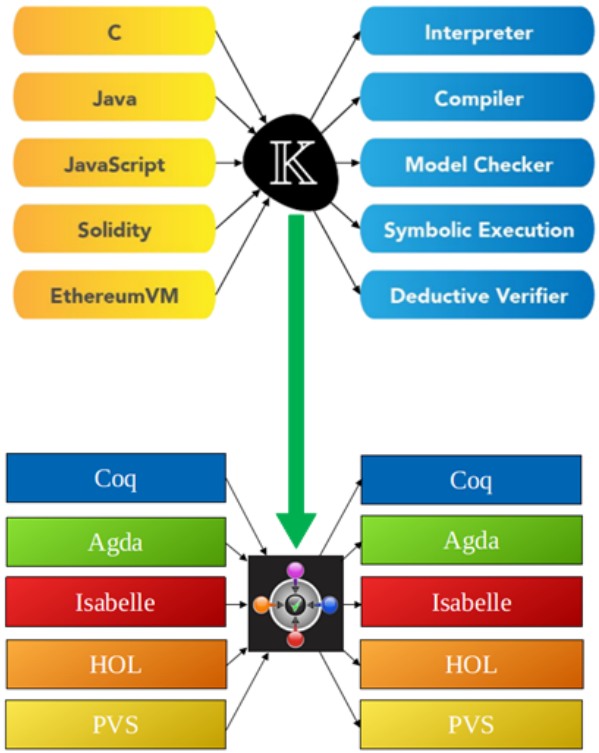
How to do that?

- **Objectives:**
  1. Formalize the translation from $\mathbb{K}$ to KORE

- **Milestones:**

How to do that?

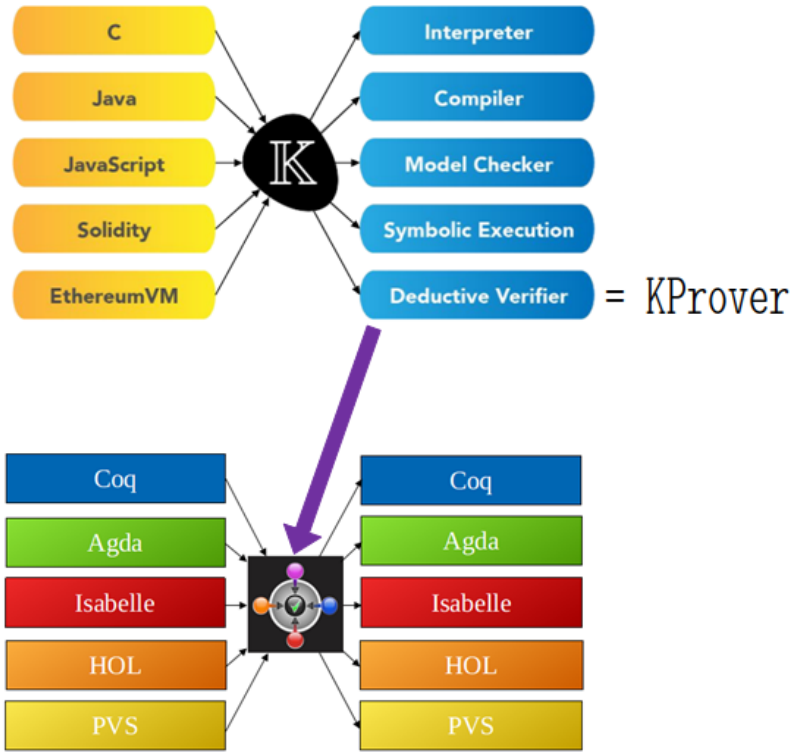
- **Objectives:**
  1. Formalize the translation from $\mathbb{K}$ to KORE
  2. Translate a $\mathbb{K}$ semantics into DEDUKTI, to execute it.



- **Milestones:**
  1. Test the semantics.

How to do that?

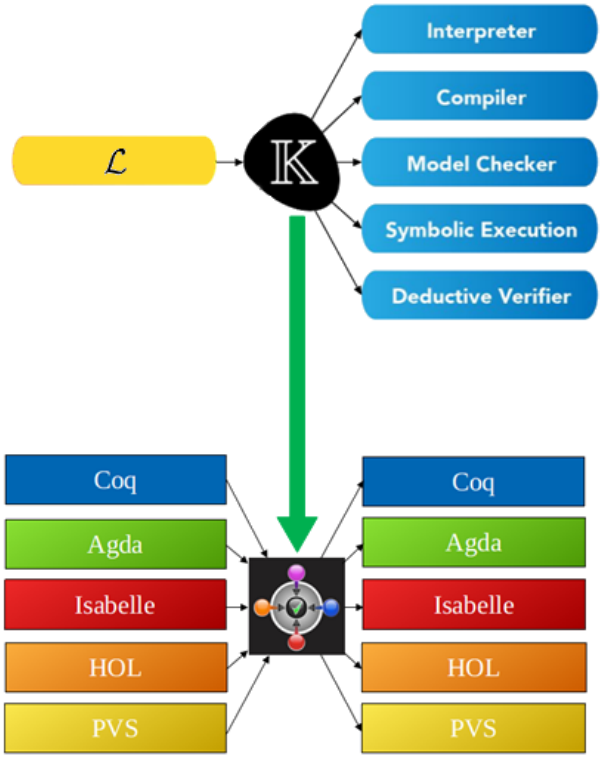
- **Objectives:**
  1. Formalize the translation from $\mathbb{K}$ to Kore
  2. Translate a $\mathbb{K}$ semantics into Dedukti, to execute it.
  3. Translate a $\mathbb{K}$ semantics into Dedukti, to check proofs.



- **Milestones:**
  1. Test the semantics.
  2. Recheck KProver proofs.

How to do that?

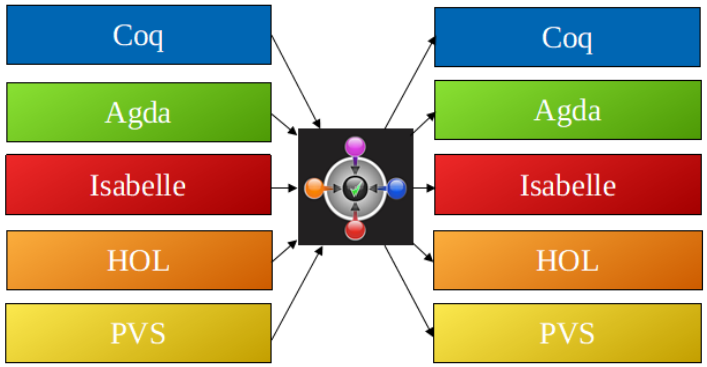
- **Objectives:**
  1. Formalize the translation from $\mathbb{K}$ to Kore
  2. Translate a $\mathbb{K}$ semantics into Dedukti, to execute it.
  3. Translate a $\mathbb{K}$ semantics into Dedukti, to check proofs.



- **Milestones:**
  1. Test the semantics.
  2. Recheck KProver proofs.
  3. Prove properties about a language $\mathcal{L}$ described in $\mathbb{K}$.

How to do that?

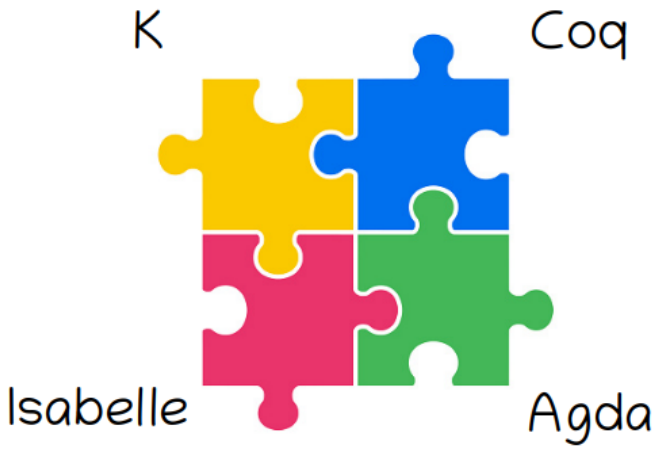
- **Objectives:**
  1. Formalize the translation from $\mathbb{K}$ to KORE
  2. Translate a $\mathbb{K}$ semantics into DEDUKTI, to execute it.
  3. Translate a $\mathbb{K}$ semantics into DEDUKTI, to check proofs.
  4. Export $\mathbb{K}$ semantics into COQ, AGDA, etc. thanks to DEDUKTI.



- **Milestones:**
  1. Test the semantics.
  2. Recheck KPROVER proofs.
  3. Prove properties about a language $\mathcal{L}$ described in $\mathbb{K}$.

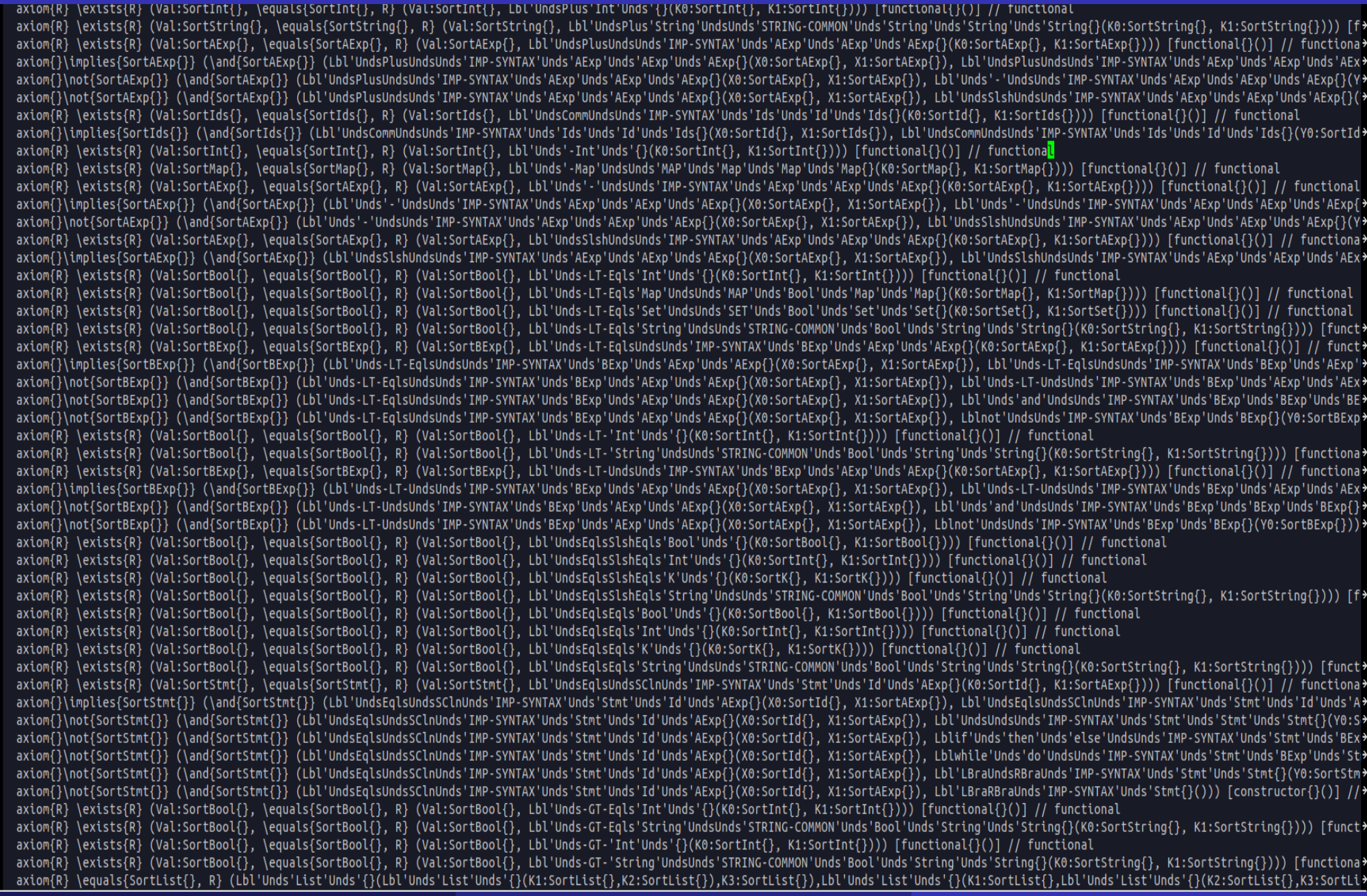
How to do that?

- **Objectives:**
  1. Formalize the translation from $\mathbb{K}$ to KORE
  2. Translate a $\mathbb{K}$ semantics into DEDUKTI, to execute it.
  3. Translate a $\mathbb{K}$ semantics into DEDUKTI, to check proofs.
  4. Export $\mathbb{K}$ semantics into COQ, AGDA, etc. thanks to DEDUKTI.

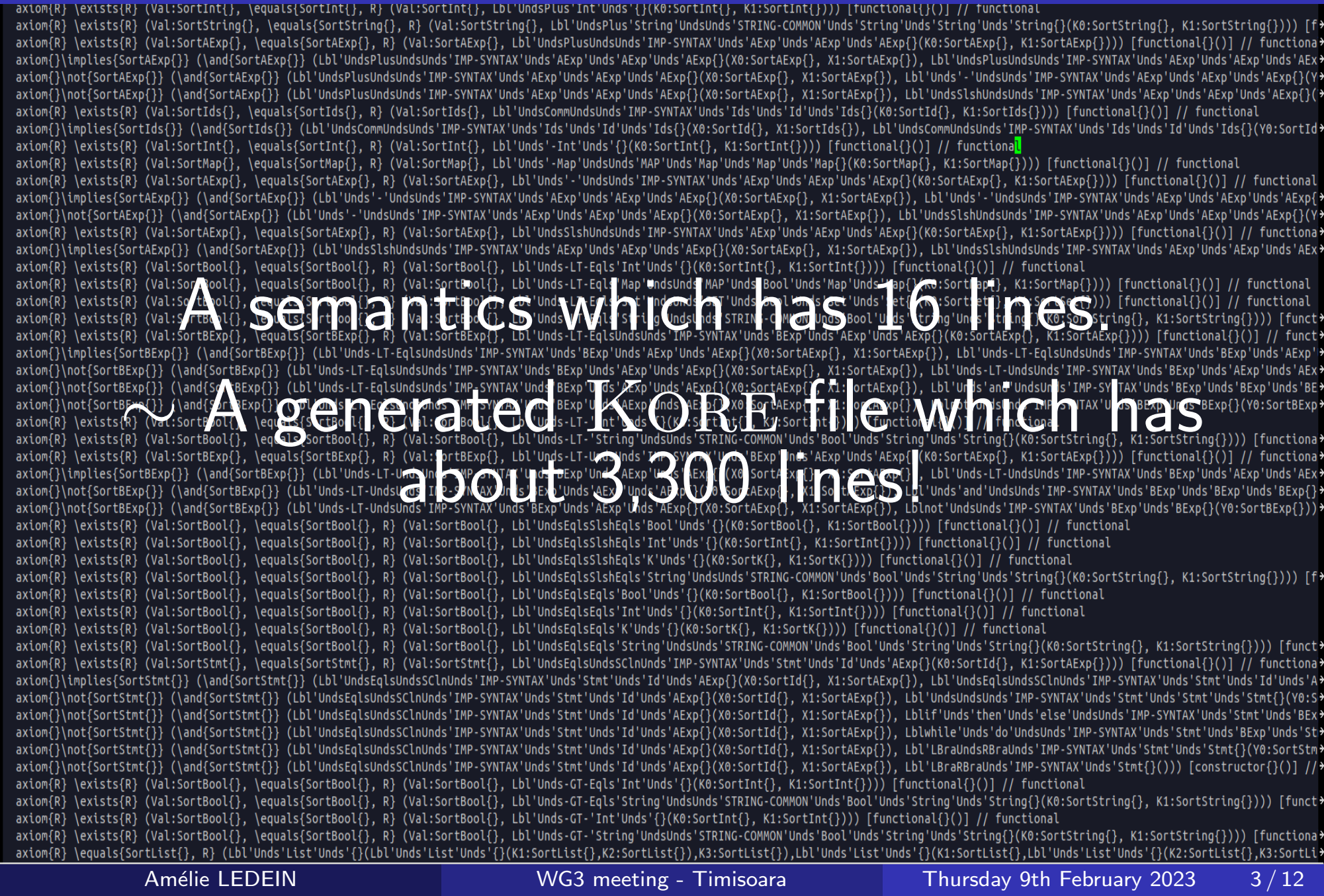


- **Milestones:**
  1. Test the semantics.
  2. Recheck KPROVER proofs.
  3. Prove properties about a language $\mathcal{L}$ described in $\mathbb{K}$.
  4. Allow multi-formalism semantics.

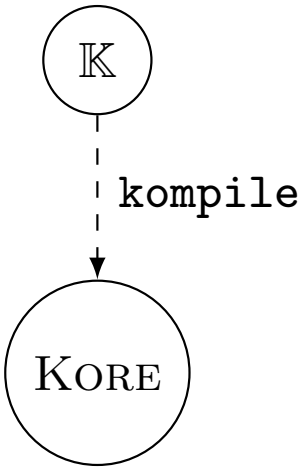
Kore: A Matching Logic theory



KORE: A MATCHING LOGIC theory

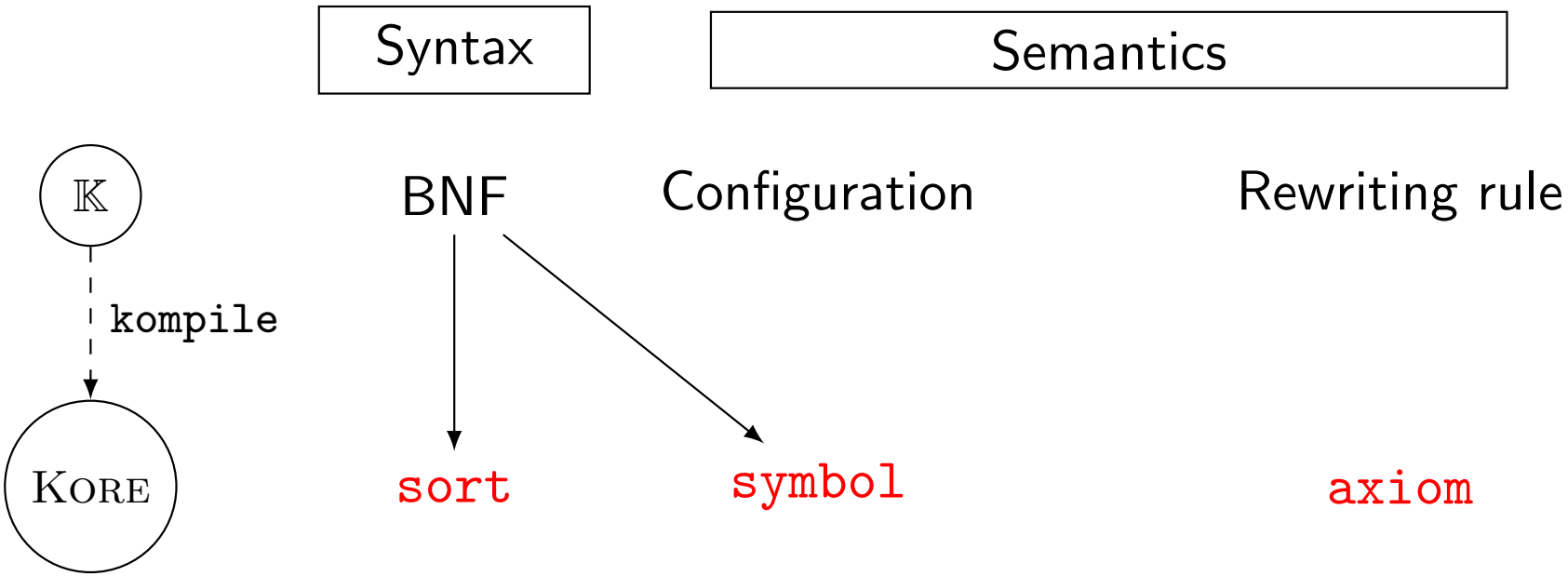


Translation from $\mathbb{K}$ to KORE

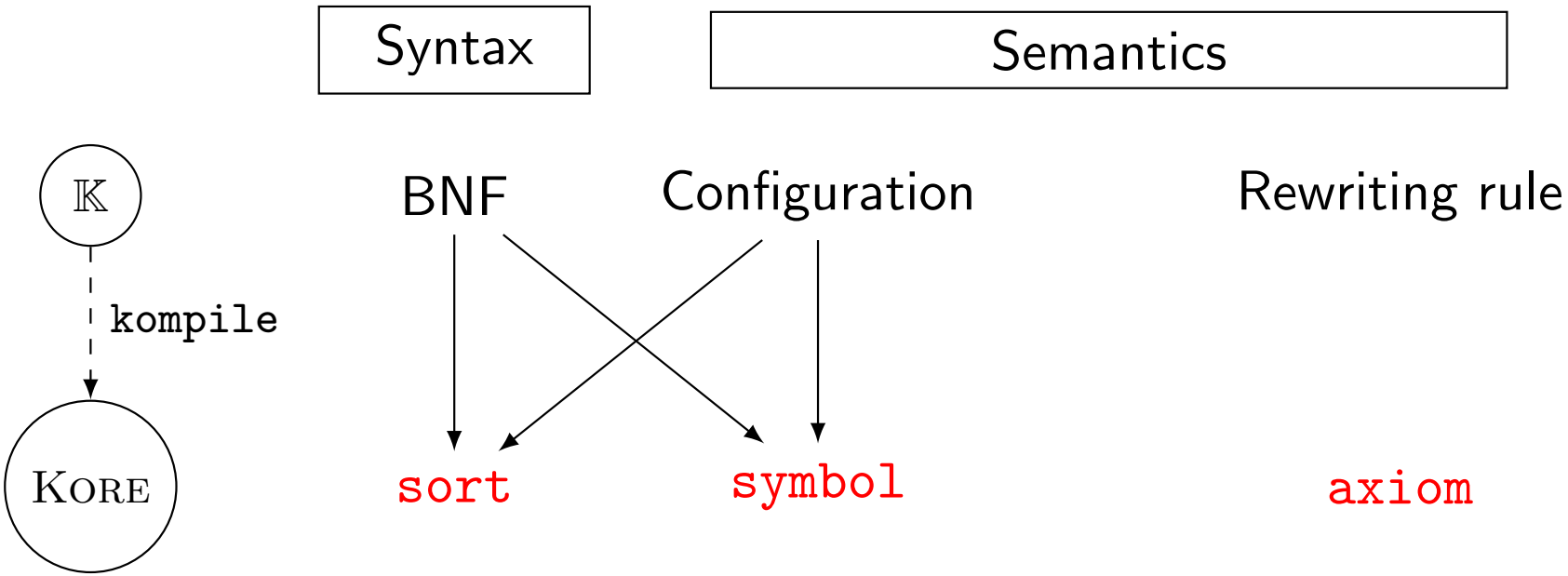


Syntax	Semantics	
BNF	Configuration	Rewriting rule
`sort`	`symbol`	`axiom`

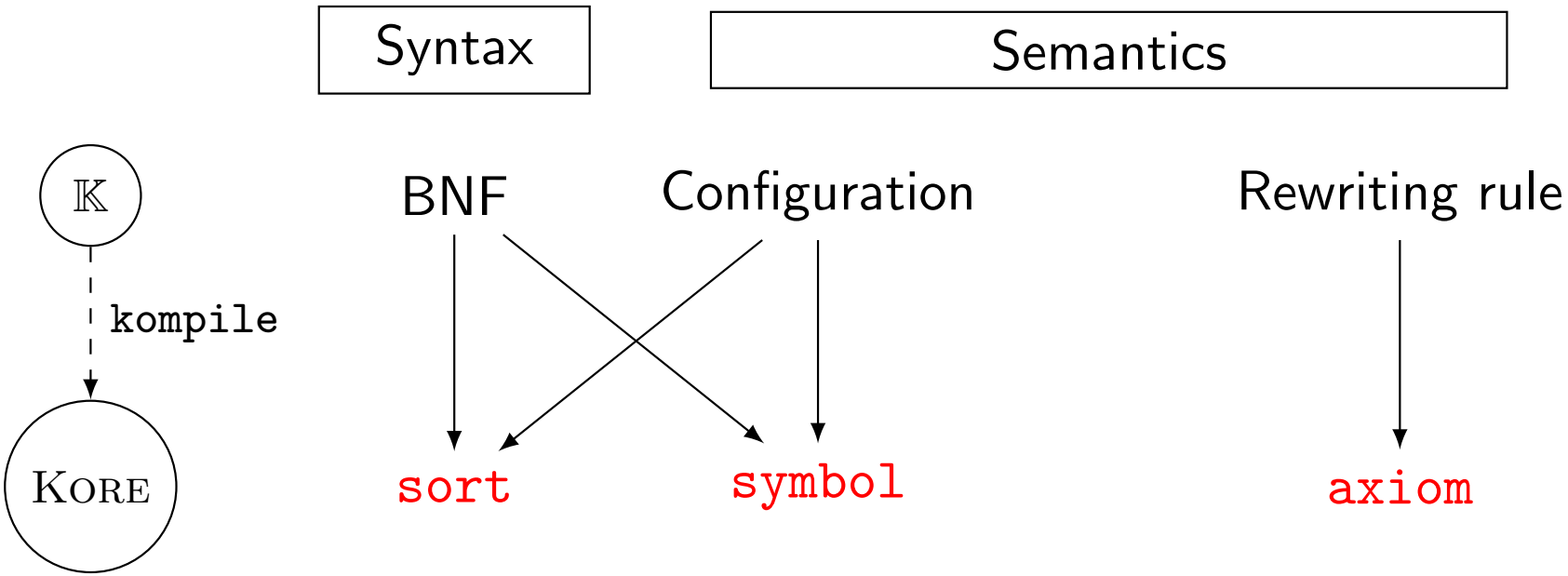
Translation from $\mathbb{K}$ to KORE



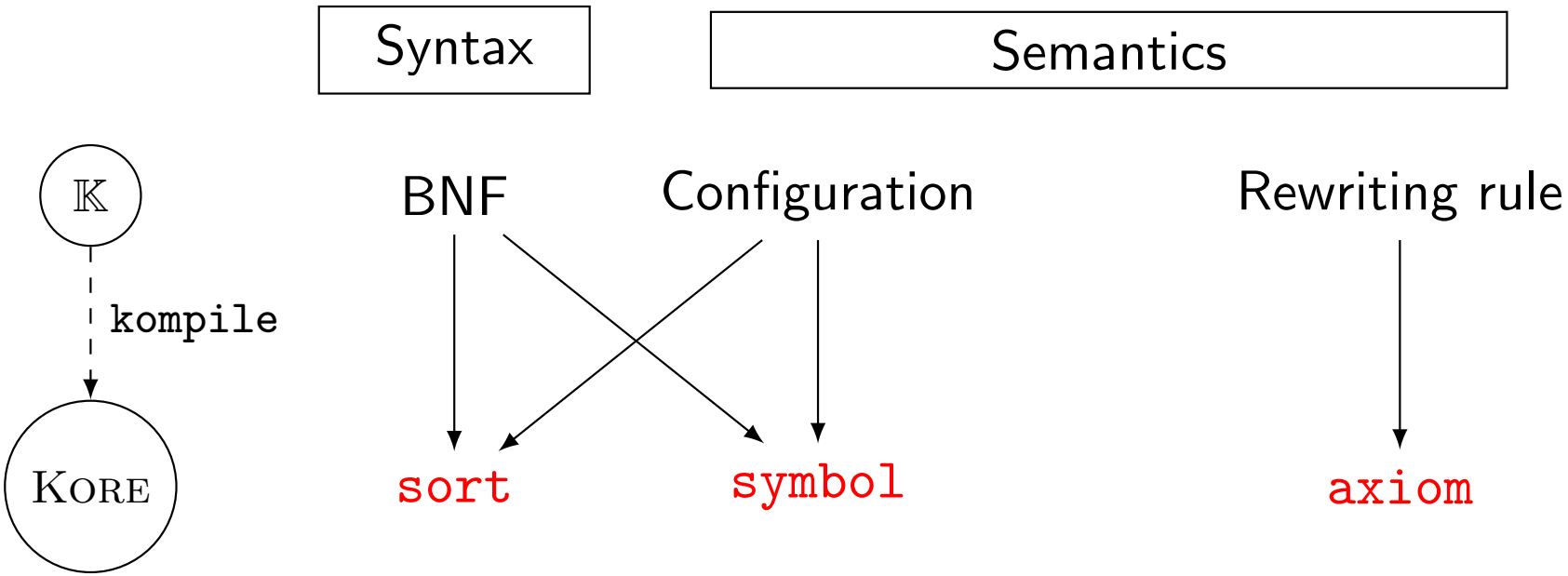
Translation from $\mathbb{K}$ to KORE



Translation from $\mathbb{K}$ to KORE



Translation from $\mathbb{K}$ to KORE



- Result: See `https://hal.science/hal-03895834/`
- Challenge: Gap between AML and μ-ML

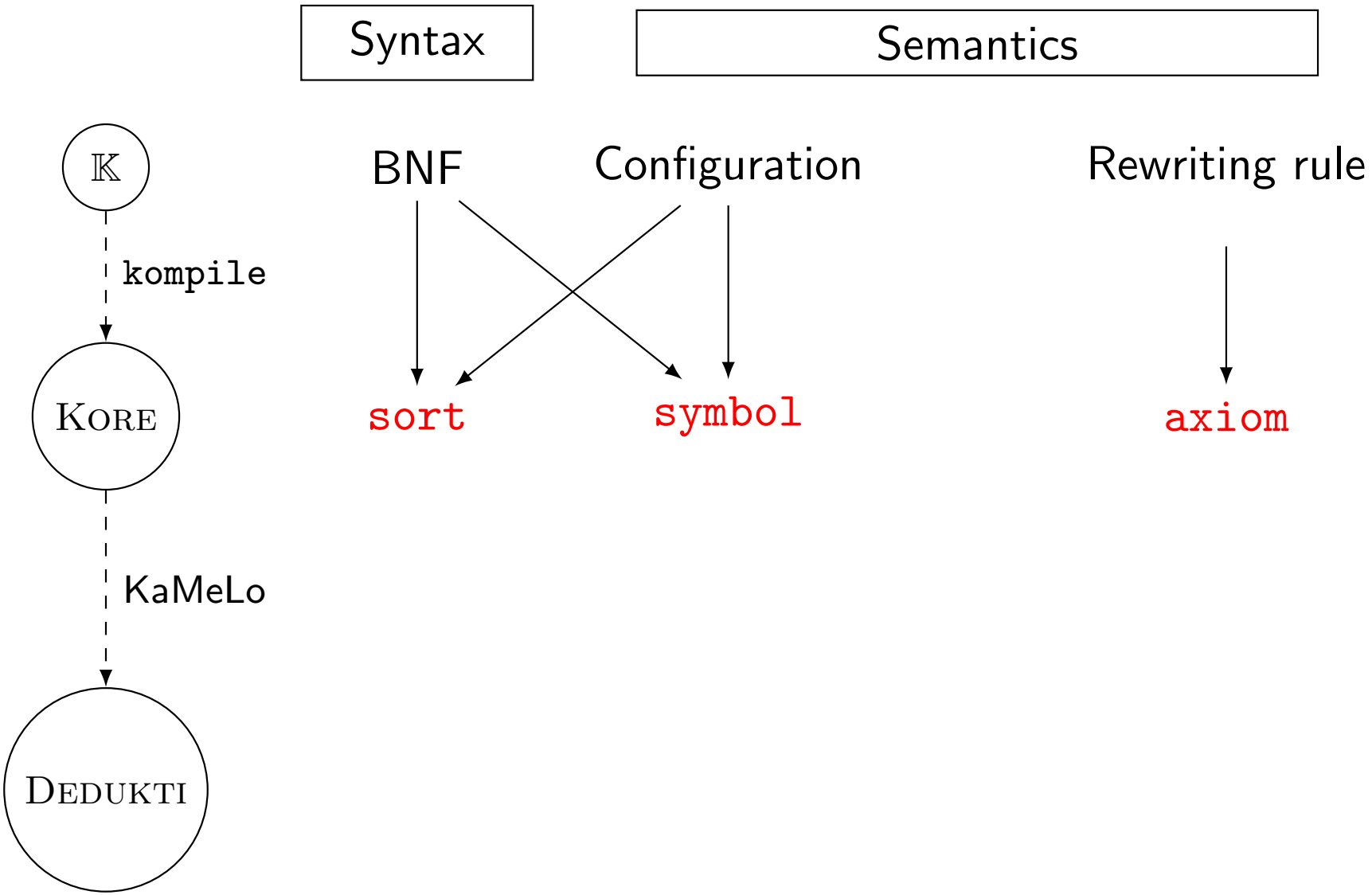
Overview of Dedukti

- Logical framework based on $\lambda\Pi$-calculus modulo theory
  $\rightarrow$ λ-calculus + dependent types + rewriting rules

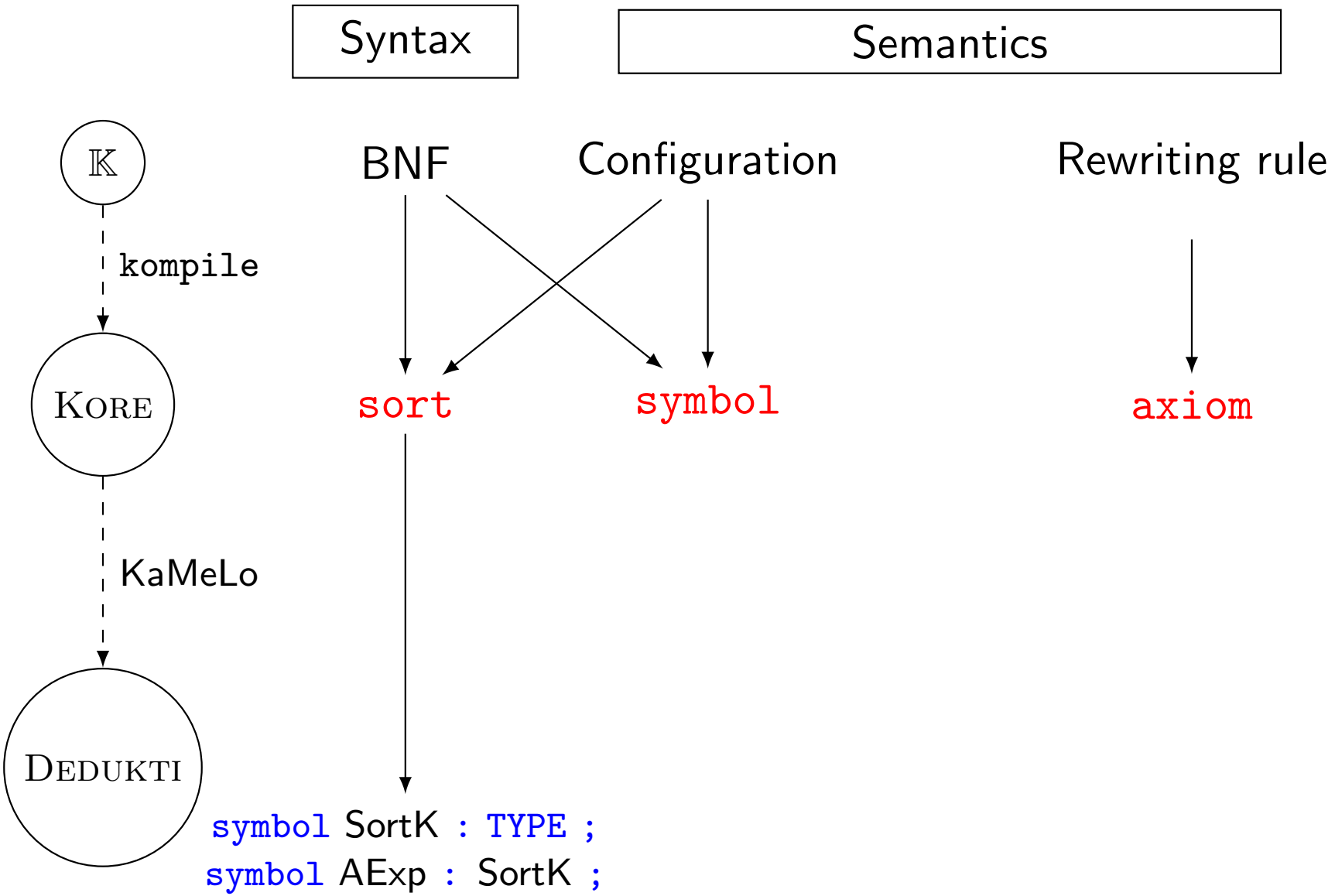
- Example of an encoding with natural numbers:

```
constant symbol Nat : TYPE;
constant symbol 0 : Nat;
constant symbol S : Nat → Nat;

symbol + : Nat → Nat → Nat;
rule $m + 0 ↪ $m;
rule $m + (S $n) ↪ S ($m + $n);

symbol add_10 : Π (_ : Nat), Nat := λ x, x + 10;
```

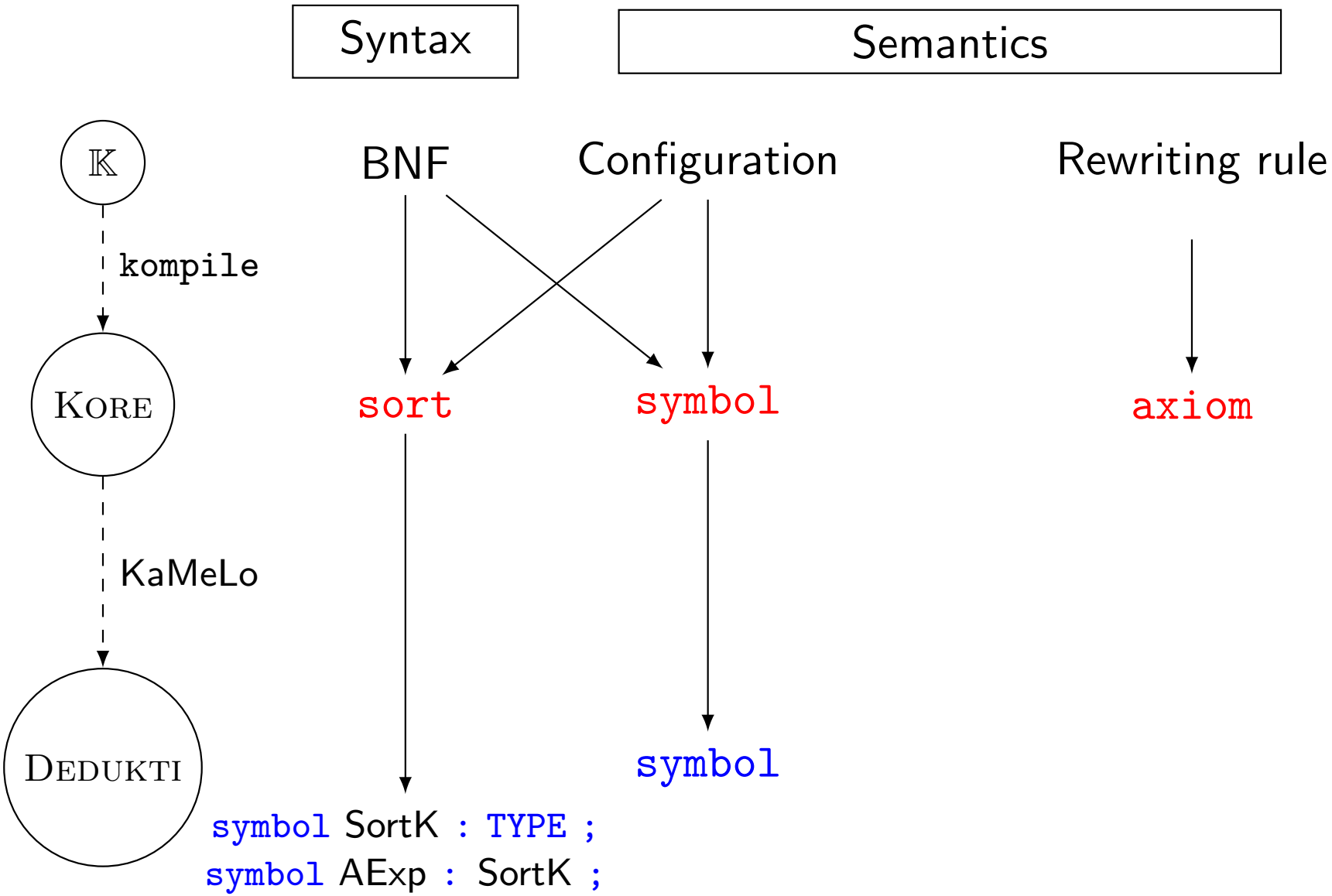
Translation from $\mathbb{K}$ to Dedukti - Execution



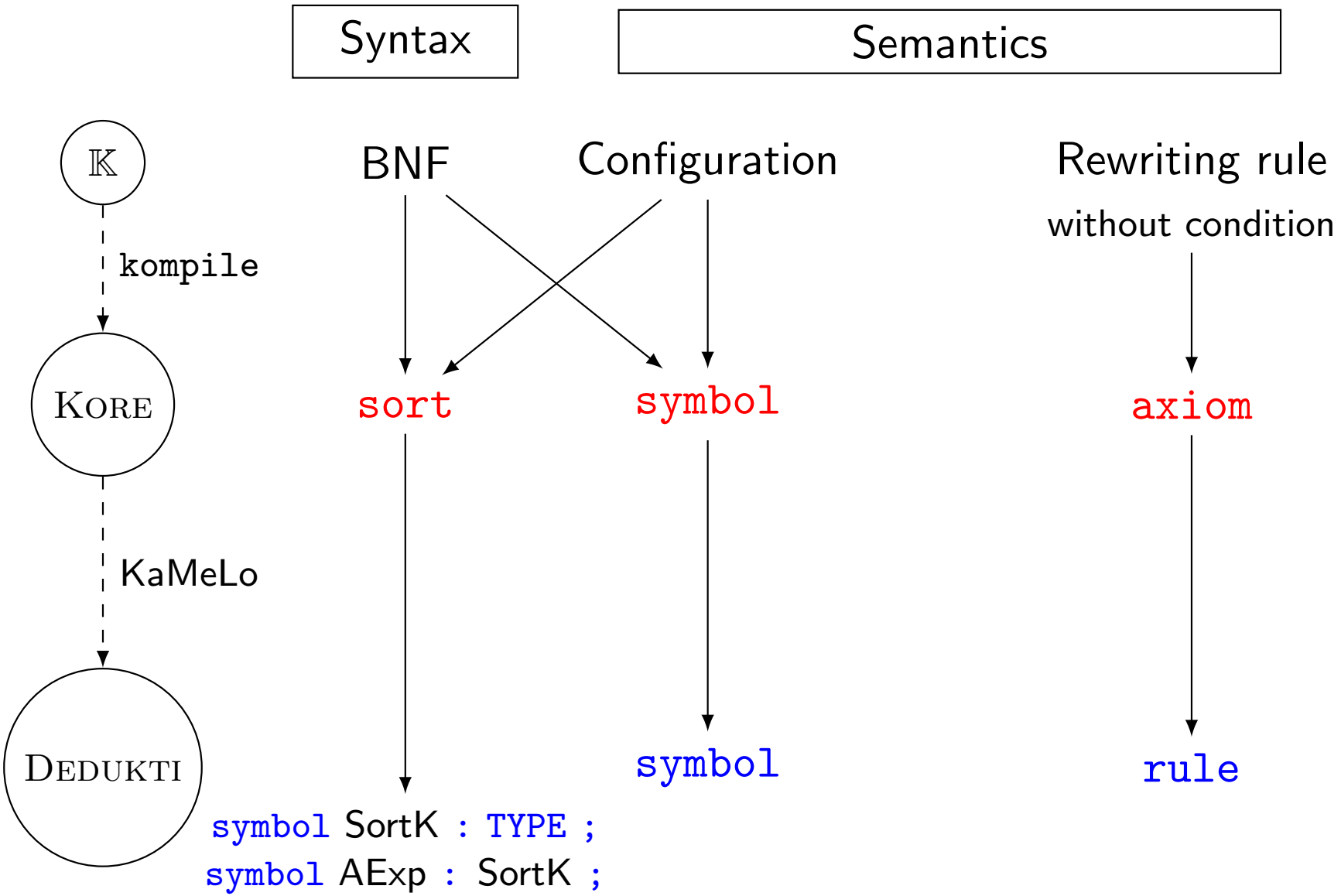
Translation from $\mathbb{K}$ to Dedukti - Execution



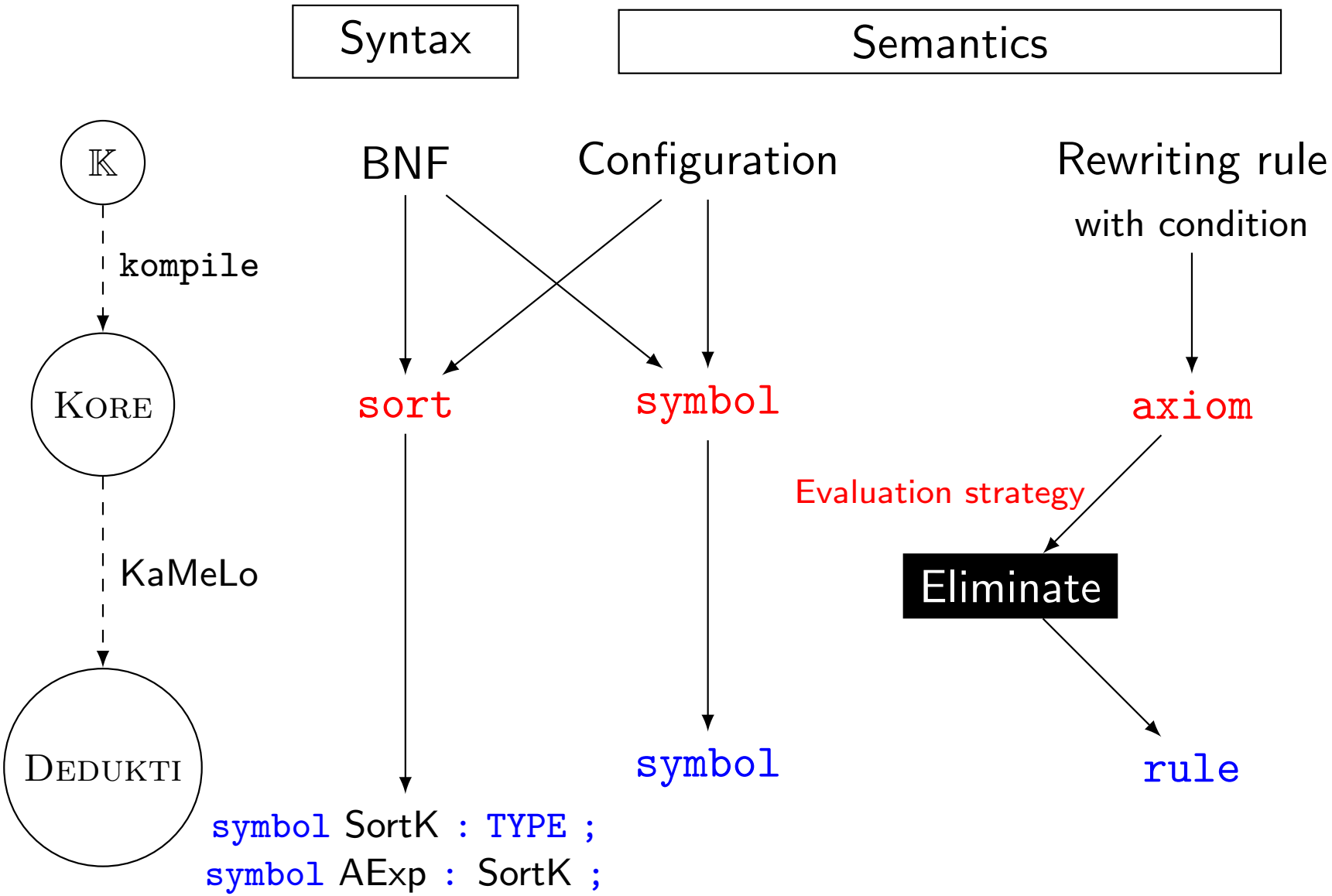
Translation from $\mathbb{K}$ to Dedukti - Execution



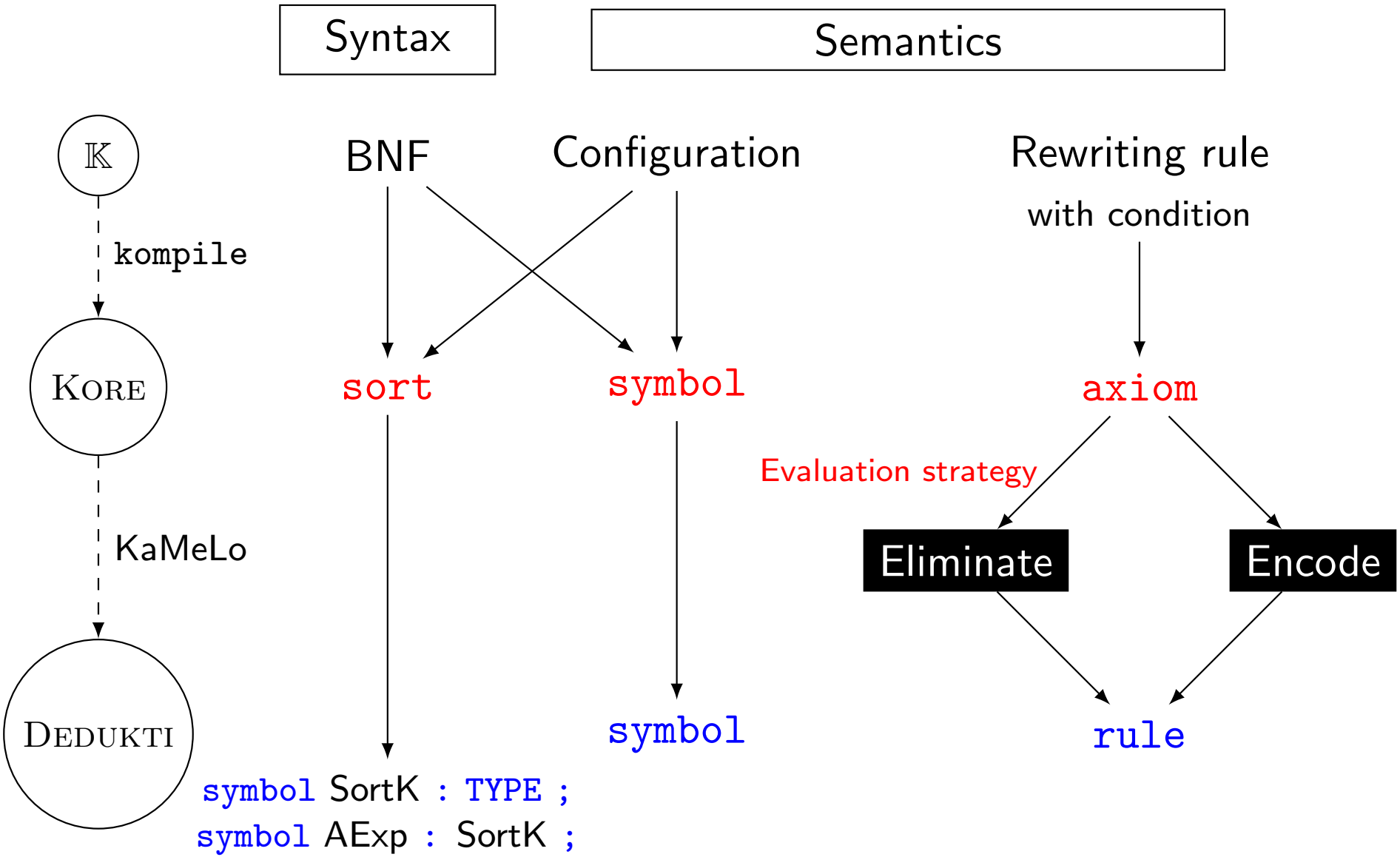
Translation from $\mathbb{K}$ to Dedukti - Execution



Translation from $\mathbb{K}$ to Dedukti - Execution



Translation from $\mathbb{K}$ to Dedukti - Execution



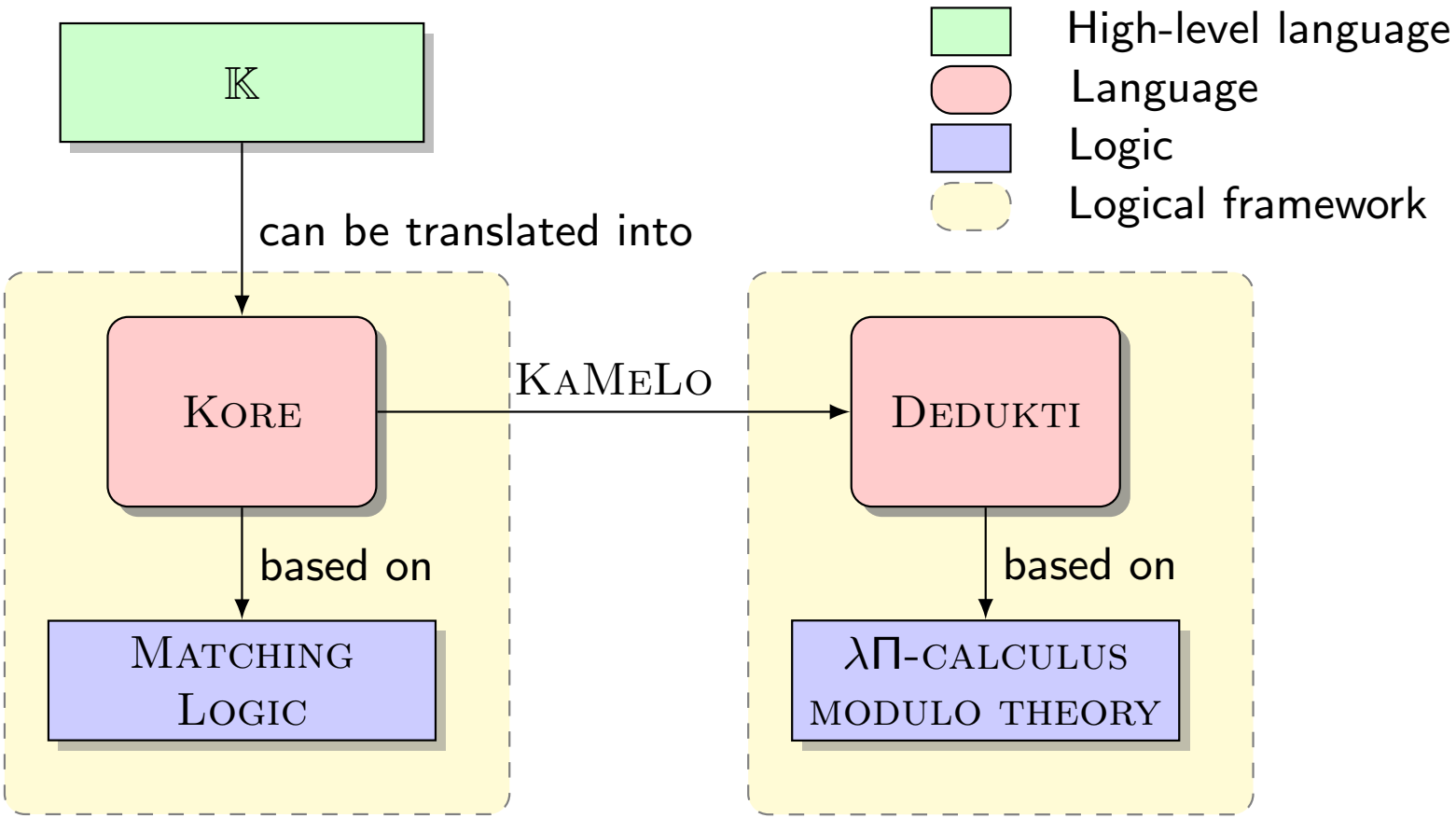
Recap

- Results
  - Implementation of a transformation[1] from CTRS to TRS
  - Automatic translator from $\mathbb{K}$ to Dedukti to execute small programs

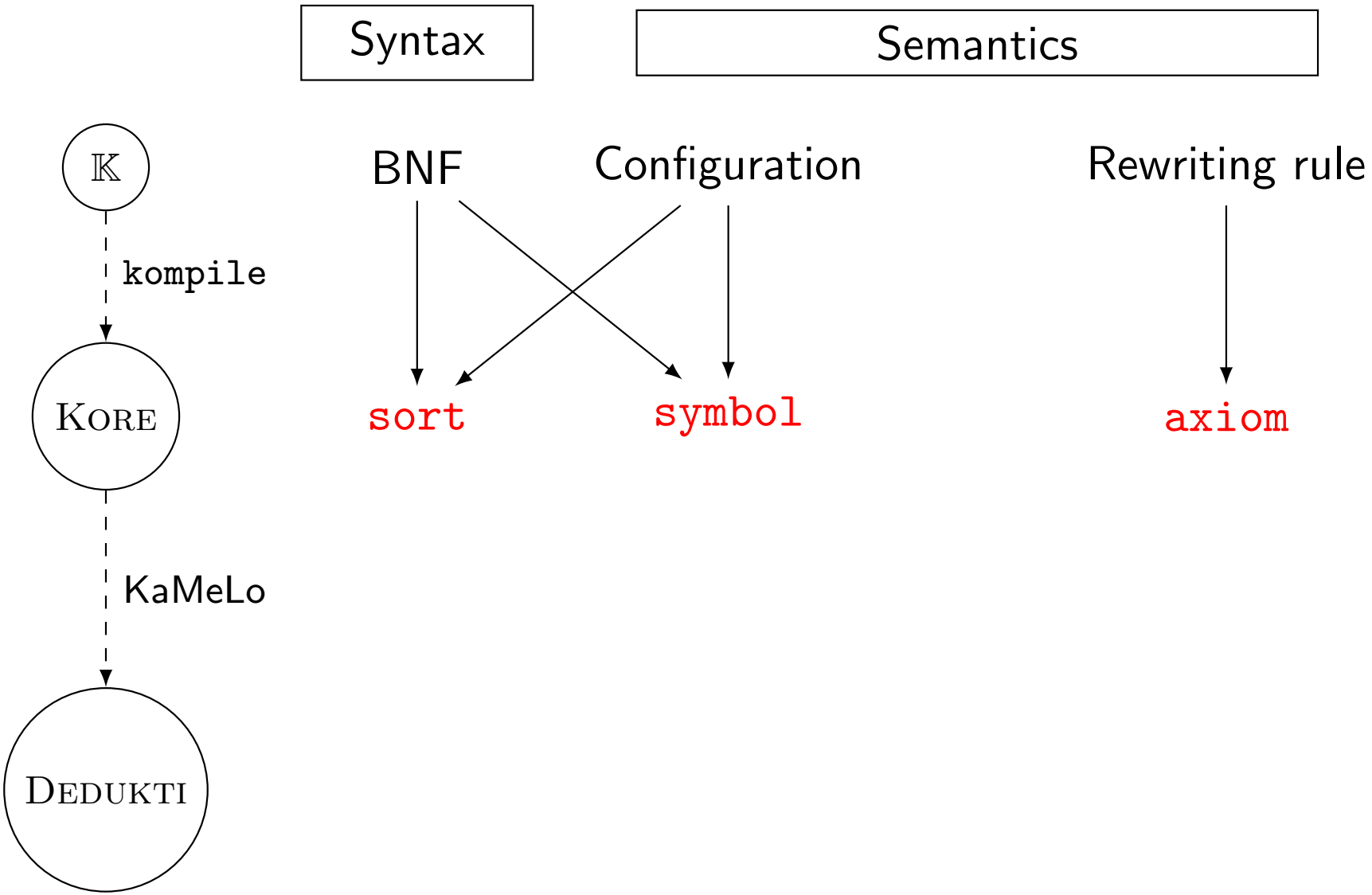
- Challenges
  - Scale up! (Threads, non-determinism, etc.)
  - Prove the preservation of confluence and termination

[1] Inspired from Patrick Viry proposal

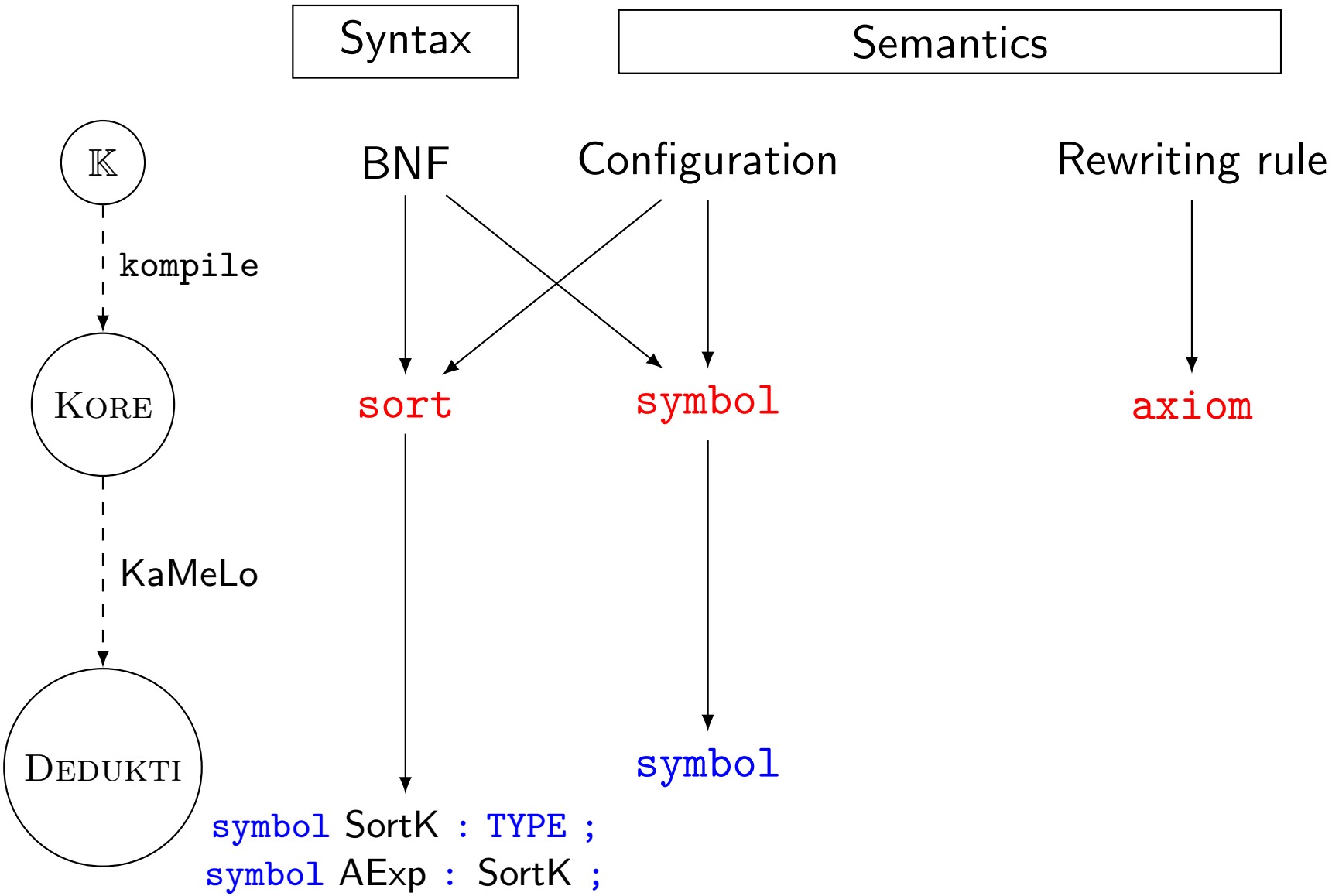
Another way to see the ecosystem



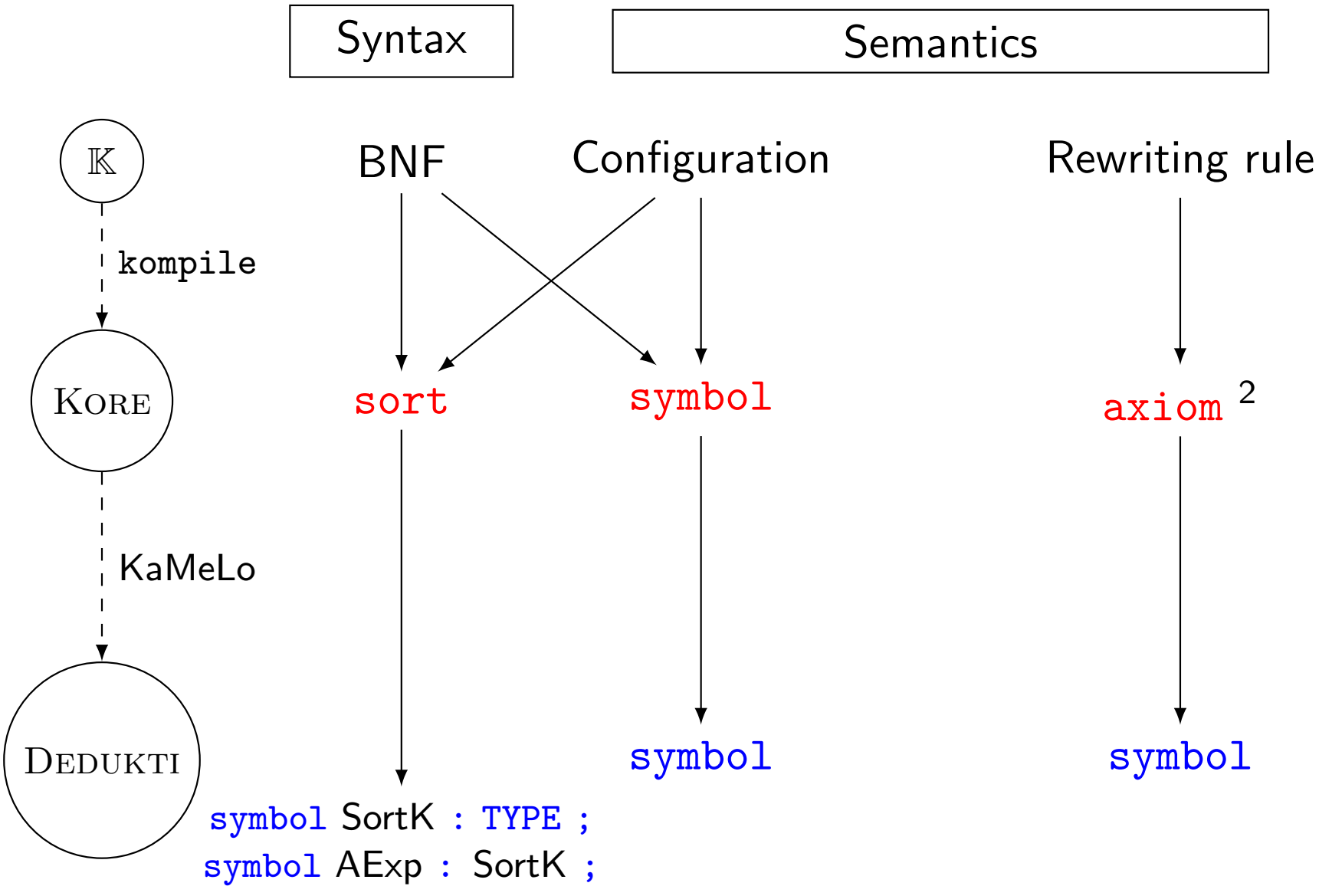
Translation from $\mathbb{K}$ to Dedukti - Verification



Translation from $\mathbb{K}$ to Dedukti - Verification



Translation from $\mathbb{K}$ to Dedukti - Verification



[2] Matching Logic pattern

A specific logic to recheck concrete executions

$$\frac{\Gamma^{\mathcal{L}} \vdash \dot{\forall}\vec{x}.\varphi}{\Gamma^{\mathcal{L}} \vdash \varphi\theta}\ \dot{\forall}\text{-elim} \qquad \varphi \in \Gamma^{\mathcal{L}}\ \frac{}{\Gamma^{\mathcal{L}} \vdash \varphi}\ \text{Axiom}$$

$$\frac{\Gamma^{\mathcal{L}} \vdash (c \mathbin{\dot{\wedge}} \varphi_A) \mathrel{\dot{\hookrightarrow}} \varphi_B \qquad \Gamma^{\mathcal{L}} \vdash c \mathrel{\dot{\succ}^*} \dot{\top}}{\Gamma^{\mathcal{L}} \vdash (\dot{\top} \mathbin{\dot{\wedge}} \varphi_A) \mathrel{\dot{\hookrightarrow}} \varphi_B}\ \dot{\hookrightarrow} \qquad \frac{\Gamma^{\mathcal{L}} \vdash c \mathrel{\dot{\to}} (\varphi_A \mathrel{\dot{\succ}} \varphi_B) \qquad \Gamma^{\mathcal{L}} \vdash c \mathrel{\dot{\succ}^*} \dot{\top}}{\Gamma^{\mathcal{L}} \vdash \dot{\top} \mathrel{\dot{\to}} (\varphi_A \mathrel{\dot{\succ}} \varphi_B)}\ \dot{\succ}$$

$$\frac{}{\Gamma^{\mathcal{L}} \vdash \varphi \mathrel{\dot{\hookrightarrow}^*} \varphi}\ \dot{\hookrightarrow}^*\text{-refl} \qquad \frac{}{\Gamma^{\mathcal{L}} \vdash \varphi \mathrel{\dot{\succ}^*} \varphi}\ \dot{\succ}^*\text{-refl}$$

$$\frac{\Gamma^{\mathcal{L}} \vdash \dot{\top} \mathrel{\dot{\to}} (\varphi_A \mathrel{\dot{\succ}} \varphi_B) \qquad \Gamma^{\mathcal{L}} \vdash \varphi_B \mathrel{\dot{\succ}^*} \varphi_C}{\Gamma^{\mathcal{L}} \vdash \varphi_A \mathrel{\dot{\succ}^*} \varphi_C}\ \dot{\succ}^*\text{-unroll}$$

$$\frac{\Gamma^{\mathcal{L}} \vdash \varphi_A \mathrel{\dot{\succ}^*} \varphi_B \qquad \Gamma^{\mathcal{L}} \vdash \varphi_B \mathrel{\dot{\hookrightarrow}^*} \varphi_C}{\Gamma^{\mathcal{L}} \vdash \varphi_A \mathrel{\dot{\hookrightarrow}^*} \varphi_C}\ \dot{\hookrightarrow}^*\text{-init}$$

$$\frac{\Gamma^{\mathcal{L}} \vdash (\dot{\top} \mathbin{\dot{\wedge}} \varphi_A) \mathrel{\dot{\hookrightarrow}} \varphi_B \qquad \Gamma^{\mathcal{L}} \vdash \varphi_B \mathrel{\dot{\succ}^*} \varphi_{B'} \qquad \Gamma^{\mathcal{L}} \vdash \varphi_{B'} \mathrel{\dot{\hookrightarrow}^*} \varphi_C}{\Gamma^{\mathcal{L}} \vdash \varphi_A \mathrel{\dot{\hookrightarrow}^*} \varphi_C}\ \dot{\hookrightarrow}^*\text{-unroll}$$

Recap

- Results
  - Partial shallow encoding of Matching Logic into Dedukti.
  - Specific Matching Logic fragment to recheck concrete KProver executions.
  - Metamath translator to Dedukti.

- Challenges
  - Symbolic execution + Circular coinduction
  - Interoperability between the Matching Logic encodings in Coq, Lean, Metamath and Dedukti

Conclusion and further works

- To recap, translators from:
  - $\mathbb{K}$ to Dedukti
  - Metamath to Dedukti
- To investigate:
  - Recheck symbolic KProver executions
  - Interoperability between the various Matching Logic encodings