UNIVERSITY OF LEICESTER

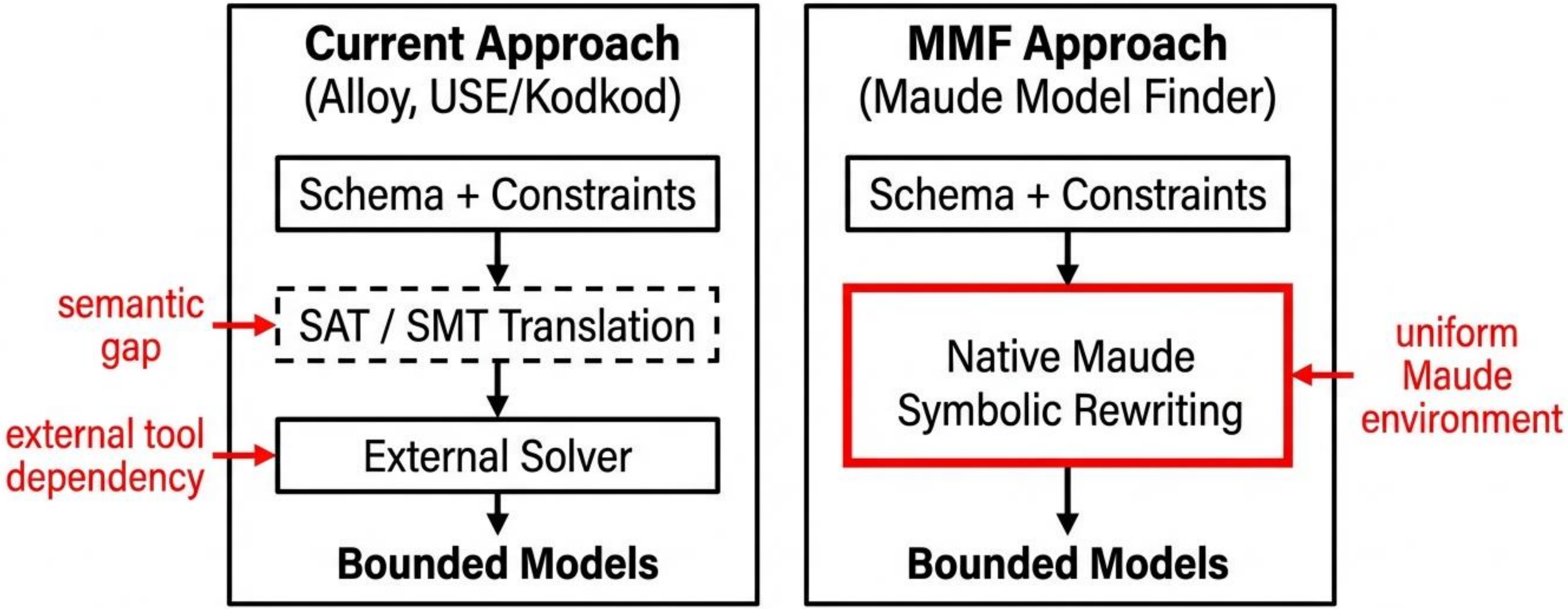
BOUNDED STRUCTURAL MODEL FINDING WITH SYMBOLIC DATA CONSTRAINTS

Artur Boronat — University of Leicester

WRLA 2026, Turin, April 2026

UNIVERSITY OF LEICESTER

MOTIVATION: THE GAP



Goal: Bounded model finding as symbolic reachability within rewriting logic

UNIVERSITY OF LEICESTER

MMF AT A GLANCE

Four Core Contributions

1 Obligation-driven two-phase calculus
Separates Object Build (`ObjBuild`) from Reference Build (`RefBuild`).

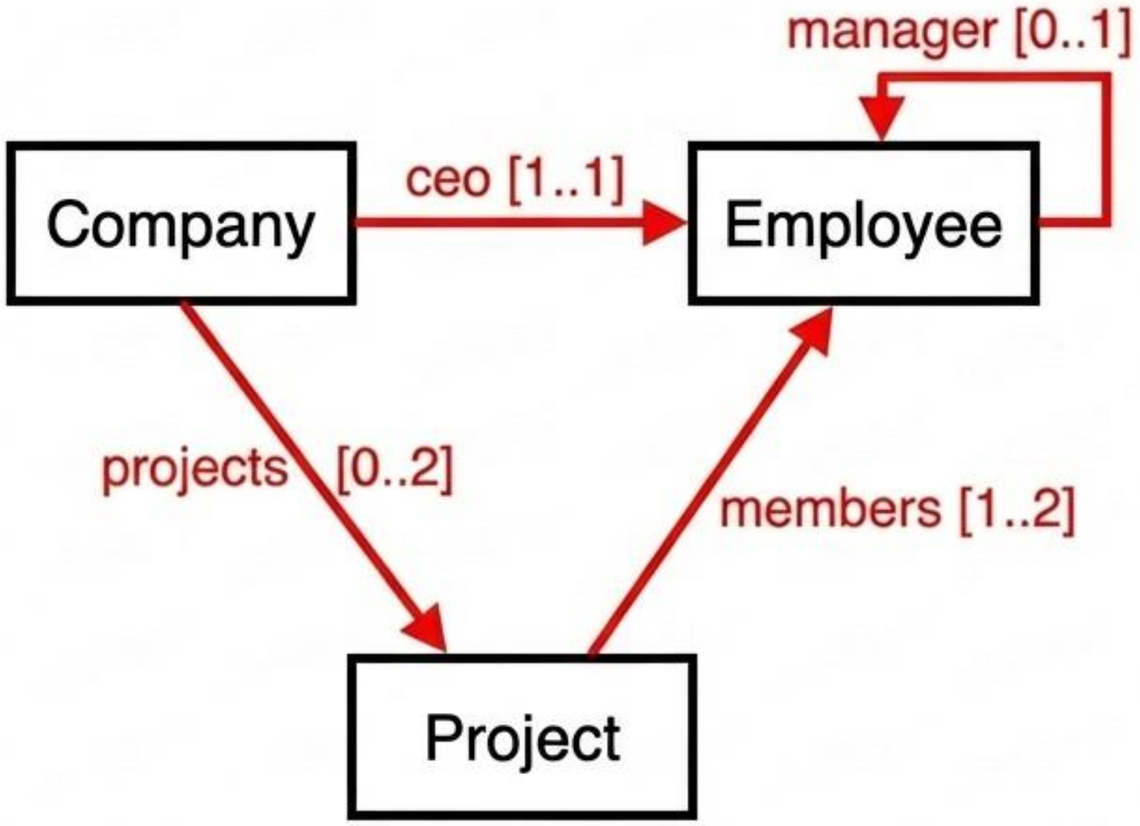
2 Hybrid SMT integration
SMT acts strictly as a background checker to prune states, not as the core solver.

3 Semantic folding
Combines shape index, ACU matching & SMT entailment.

4 Formal guarantees
Ensures termination, soundness, and bounded completeness.

RUNNING EXAMPLE: SCHEMA

Class Diagram Topology



Maude Schema & Bounds

```
class Company | ceo : Oid, projects : Set{Oid} .

class Project | members : NeSet{Oid} .

class Employee | manager : Maybe{Oid} .
```

```
eq boundsDecl =
  classB(Company, 1, 1) ;
  classB(Employee, 2, 2) ;
  classB(Project, 0, 2) ;
  roleB(Company, ceo, Employee, 1, 1) ;
  roleB(Employee, manager, Employee, 0, 1)
  ... .
```

CONSTRAINTS

UNIVERSITY OF LEICESTER

Structural Constraints (Graph)

Example: Acyclic management & CEO rules

```
eq specErrorPredGraph(CF) =
    ceoHasManager(CF)
    ttor
    managerCyclic(CF).
```

- Evaluated as truth predicates (tt)
- Can be monotonic or non-monotonic

Data Constraints (Symbolic)

Example: Valid integer levels for rank

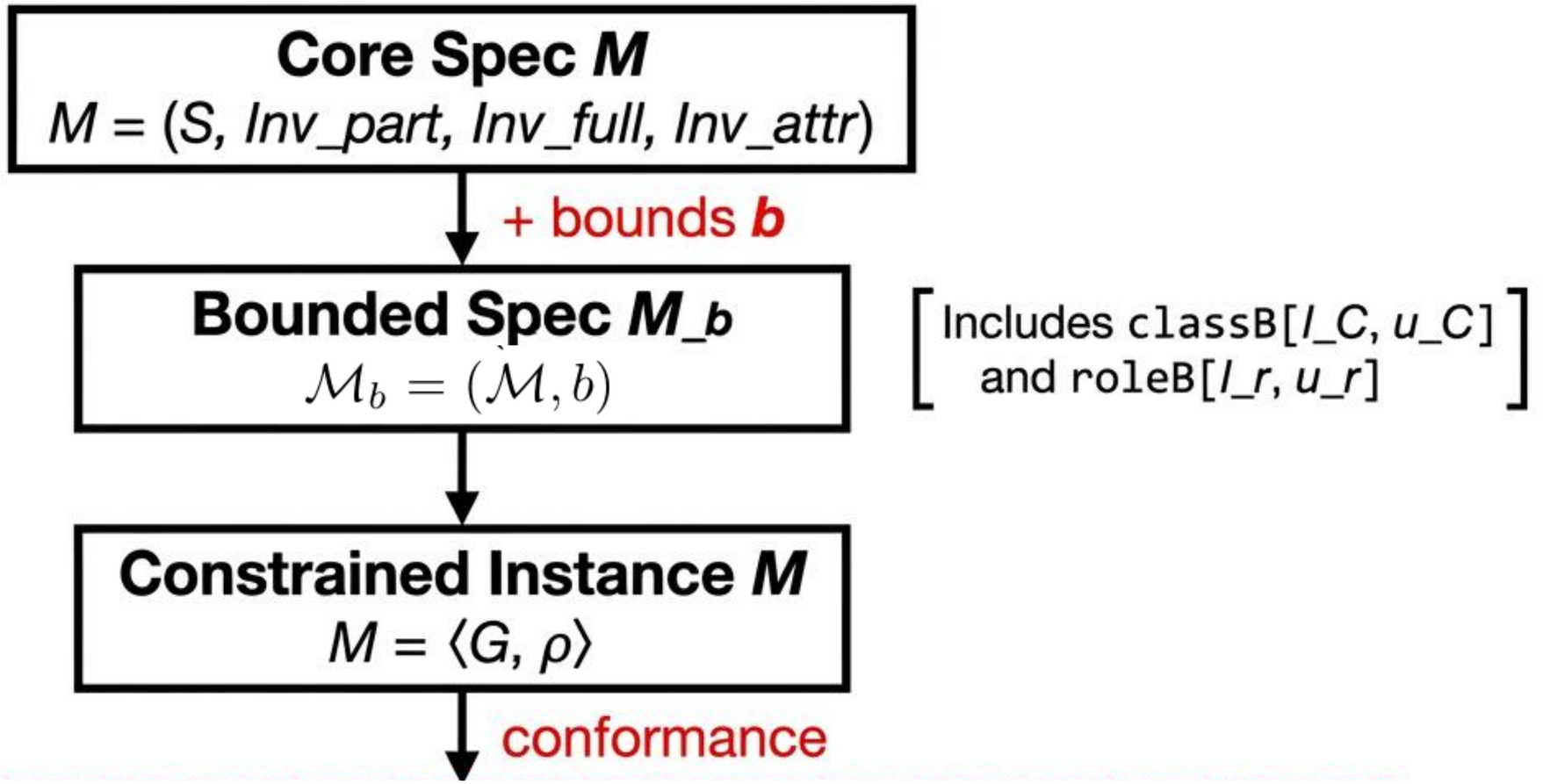
```
eq specInvOnCreate(Employee,E) =
   i(E,lvl) >= 0 and i(E,lvl) < 3 .
ceq specInvOnSetRef(Emp,mgr,E,M) =
   i(M,lvl) < i(E,lvl) if M :: Oid.
eq specInvOnSetRef(Comp,ceo,C,E) =
   i(E,lvl) === 0 .
```

- Solved in background via SMT variables

Key Execution Concept

Monotonic partial preds -> **Early pruning** | Non-monotonic full preds -> **Normal Forms only**

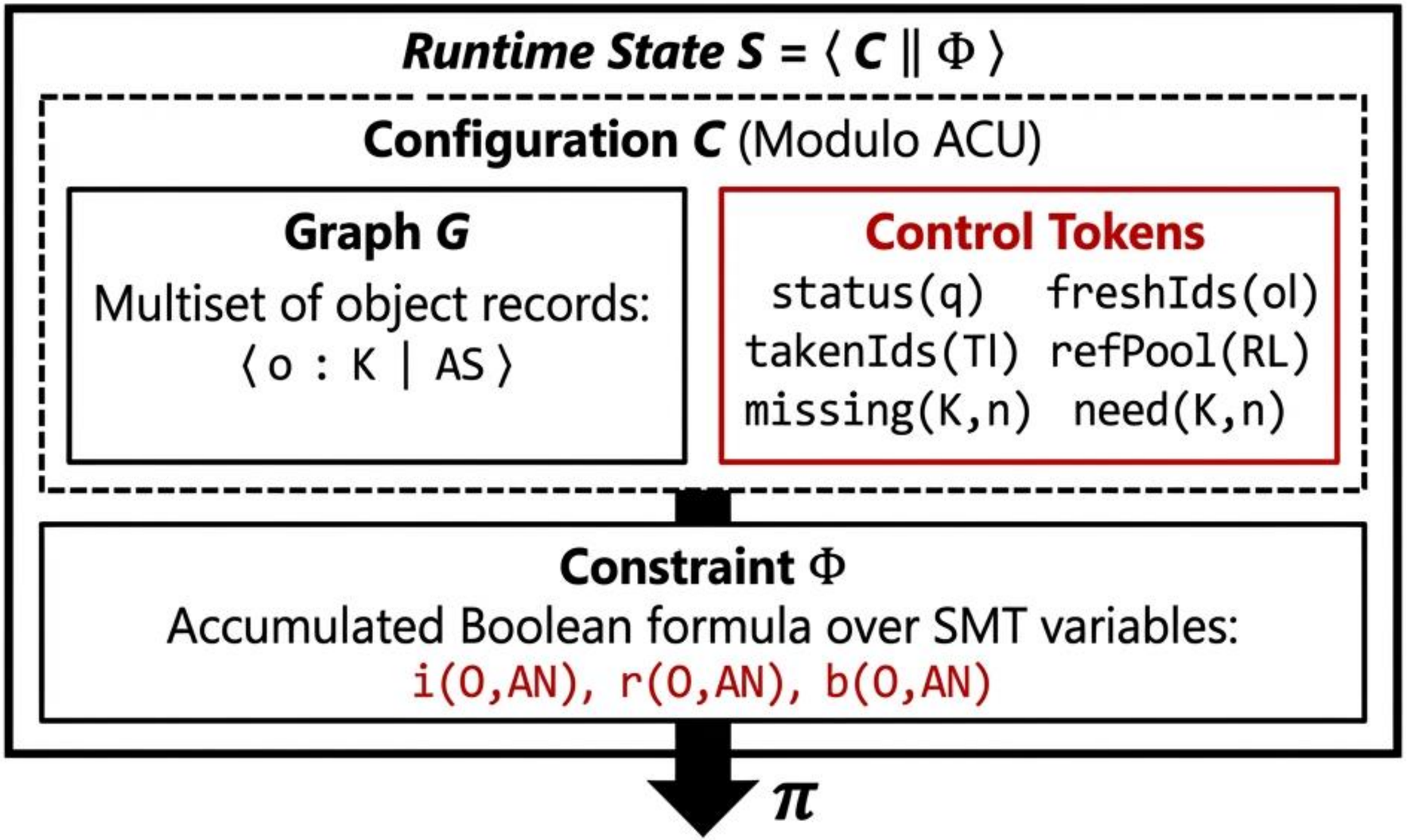
SEMANTIC FOUNDATIONS



$$G : \mathcal{S}_b \;\wedge\; (G \models \mathsf{Inv}^{\mathsf{part}}_{\mathsf{graph}}) \;\wedge\; (G \models \mathsf{Inv}^{\mathsf{full}}_{\mathsf{graph}}) \;\wedge\; (\rho \models \mathsf{Inv}_{\mathsf{attr}}(G))$$

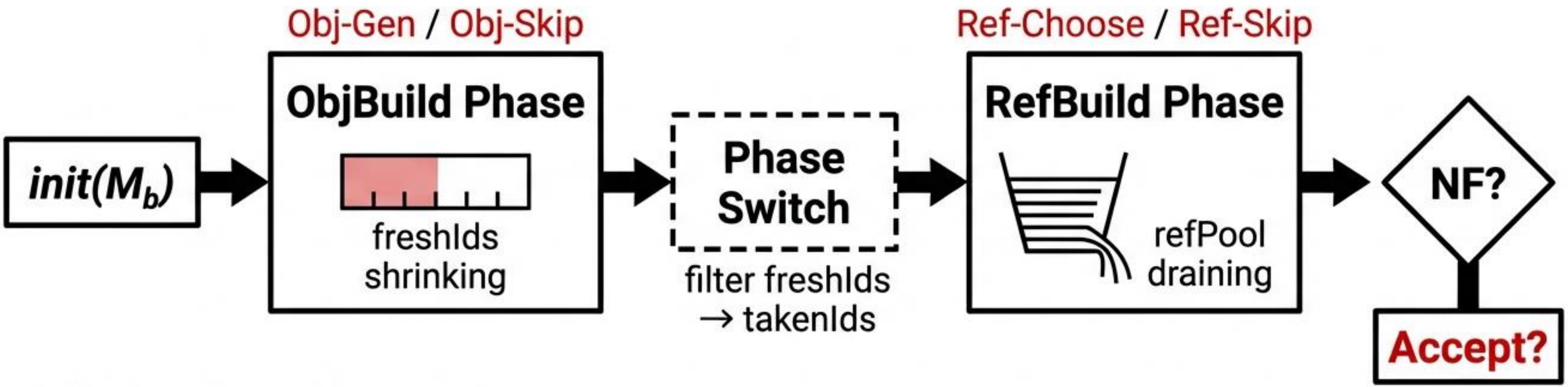
$$[\![\mathcal{M}_b]\!] \;=\; \{\, M \mid M : \mathcal{M}_b \,\}$$

RUNTIME STATES



$\pi(C)$ erases control tokens; $[\![S]\!] = \{ \langle \pi(C), \rho \rangle \mid \rho \vDash \Phi \}$

TWO-PHASE CALCULUS OVERVIEW



Calculus Step Semantics:

1. Raw	$C \longrightarrow C_1$	(metaApply **mod** B)
2. Normalized	$C_1 \Rightarrow C'$	(metaReduce E/B)
3. Constrained	$\langle C' \parallel \Phi' \rangle$	Φ' = simpB(Φ **AND** hook(σ))

OBJBUILD RULES

UNIVERSITY OF LEICESTER

Obj-Gen

Canonical allocation order

$$\frac{\texttt{freshIds}(o_K :: ol),\ \texttt{missing}(K,N),\ \texttt{need}(K,N')\ \ldots \parallel \Phi}{\texttt{freshIds}(ol),\ \texttt{missing}(K,N-1),\ \texttt{allocate}\ \langle o_K{:}K|\ldots\rangle,\ \texttt{enqueue refPool}\ \ldots \parallel \Phi'}$$

Obligation tracking

$\Phi' = \texttt{simpB}(\Phi \wedge \texttt{hook\_create}(K, o_K))$

Obj-Skip

$$\langle\ \texttt{freshIds}(o_K :: ol),\ \texttt{missing}(K, 0),\ \texttt{need}(K, 0)\ \ldots \parallel \Phi\ \rangle$$
$$\Rightarrow \langle\ \texttt{freshIds}(ol),\ \texttt{missing}(K, 0),\ \texttt{need}(K, 0)\ \ldots \parallel \Phi\ \rangle$$

**Only when lower bounds met*

REFBUILD RULES

Ref-Choose

$$\frac{\texttt{refPool(refIdx(…, curr, REM) :: RTL) …} \parallel \Phi}{\texttt{refPool(RTL), assign } \mathbf{v = choiceRef(…)} \texttt{ …} \parallel \Phi'}$$

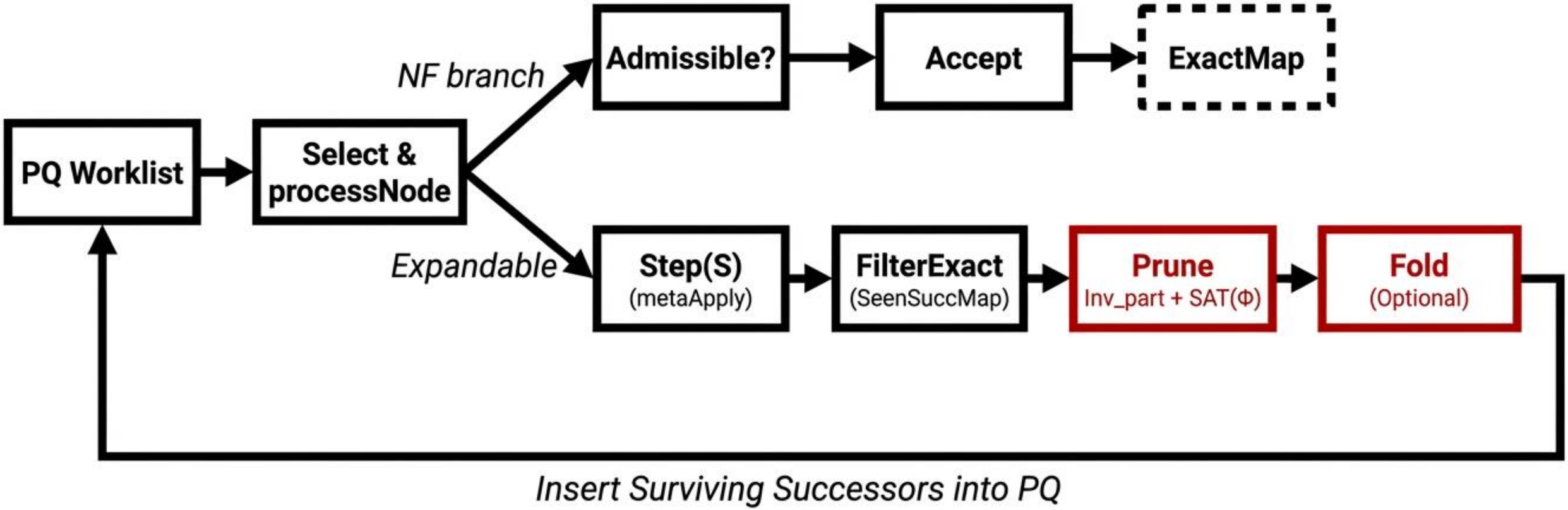
v = unrank(P, m, curr) → stateless extraction of j-th subset
Φ' includes **hook_setref**

Ref-Skip

$$\langle\ \texttt{refIdx(…, curr, REM) :: RTL …}\ \rangle$$
$$\Rightarrow \langle\ \texttt{refIdx(…, curr+1, REM-1) :: RTL …}\ \rangle$$

**Shrinking rank interval REM-1 ensures termination*

REACHABILITY ENGINE




```
Loop state L = (PQ, SeenMap, SeenSuccMap, ExactMap, Profiler)
```

SUBSUMPTION AND FOLDING

UNIVERSITY OF LEICESTER

Semantic Coverage Preorder Gates ($S_1 \leq S_2$)

Gate 1: Shape Index

$\eta(S_1) = \eta(S_2)$

Performance filter

→

Gate 2: Structural Embedding

metaMatch: $\pi(C_2) =_{E+B} \pi(C_1)\sigma$

Embedding mod ACU

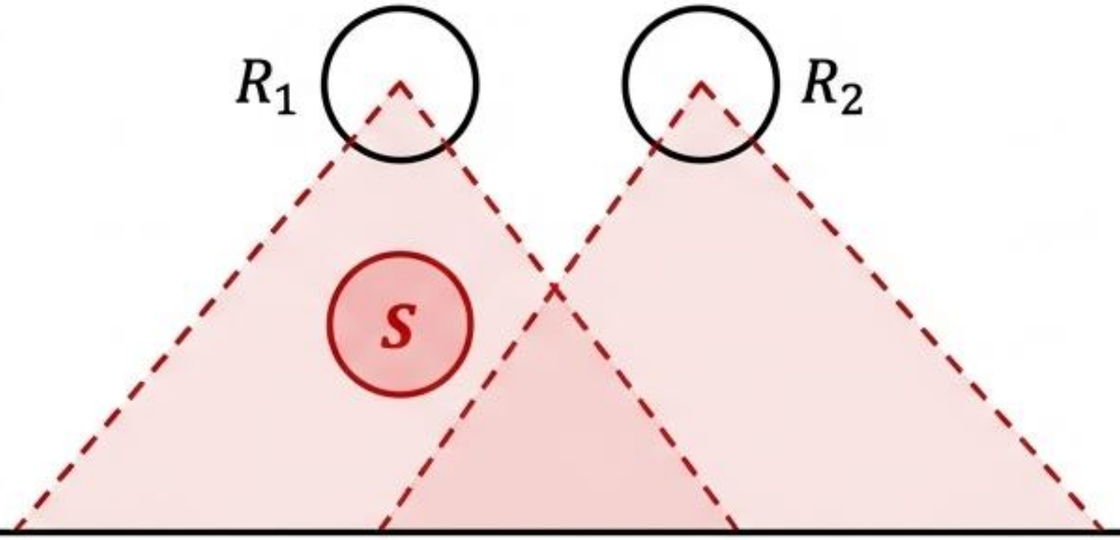
→

Gate 3: SMT Solver

$\Phi_2 \models \Phi_1\sigma$

Constraint entailment

Covering Antichain A



FoldSift (Queue-time pruning)

Discard S if subsumed by representative $R \in A$.

FoldRefresh (Repository maintenance)

Evict R if new state S supersedes it.

⇒ **Ensures Coverage-Preserving Search**

UNIVERSITY OF LEICESTER

CORRECTNESS

Termination

Lexicographic measure **strictly** decreasing, bounded **by scope**:

$(|\texttt{freshIds}|, \Sigma(\texttt{REM}))$

Soundness

$\texttt{Accept\_find(S)} \Rightarrow [[\texttt{S}]] \subseteq [[\texttt{M\_b}]]$

Every reported model conforms **strictly** to the specification.

Completeness

$\forall\, \texttt{M} \in [[\texttt{M\_b}]], \exists\, \textbf{accepted}\ \texttt{S} : \texttt{M} \in [[\texttt{S}]]$

No valid model missed within the declared bounds.

Folding Soundness

Antichain A is coverage-preserving:

$\forall\, \textbf{discarded}\ \texttt{S}, \exists\, \texttt{R} \in \texttt{A} : [[\texttt{S}]] \subseteq [[\texttt{R}]]$

Full proofs in extended version

UNIVERSITY OF LEICESTER

EVALUATION: MODES AND ABLATIONS

Mode comparison (1 Co, 2 Emp, 0-2 Proj)

Mode	ms	rew(M)	pop	pruned
findAll(find)	2181	7.92	450	354
findFirst(find)	136	0.28	29	27
findAll(check)	2409	11.1	374	298
findFirst(chk)	144	0.31	29	21

Structural pruning is largest contributor

Ablations (findAll(find))

Config	ms	pop	NF
baseline	2185	450	52
noSMT	TIMEOUT	-	-
noFolding	1258	494	96
folding-noIndex	TIMEOUT	-	-
noSMT-noFolding	20299	2054	192

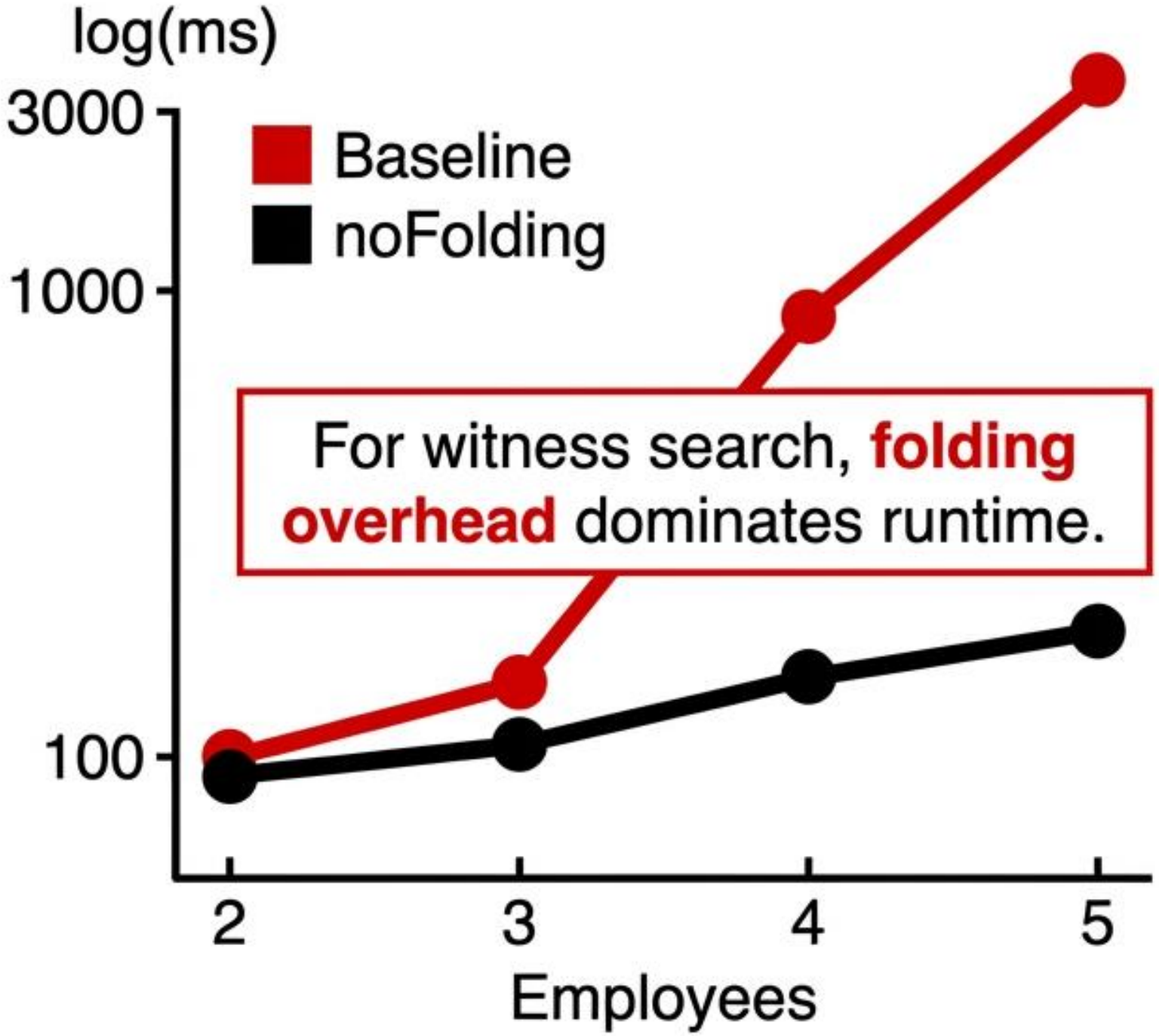
SMT pruning essential

Folding requires shape indexing

SCALABILITY

findFirst(check) scaling

Emp	Baseline	noFold
2	**97 ms**	88 ms
3	**147 ms**	107 ms
4	**737 ms**	172 ms
5	**3098 ms**	264 ms



UNIVERSITY OF LEICESTER

RELATED WORK

Reuses: constrained rewriting, metaMatch, folding

SAT/SMT Finders

- **Alloy, USE/Kodkod**
- Translation-based
- Semantic gap limits native integration
- External solver dependencies

Maude Symbolic

- **Bae & Meseguer:** Reachability mod SMT
- **Escobar et al.:** Folding for narrowing
- **Arias et al.:** Constrained subsumption

MMF (This Work)

Novel additions:

- Two-phase calculus
- Ranked enumeration
- Shape indexing

- **Native Maude**
- **Hybrid SMT**

UNIVERSITY OF LEICESTER

CONCLUSION AND FUTURE WORK

Contributions

1. Two-phase obligation-driven construction calculus
2. Hybrid SMT leveraged strictly as a background checker
3. Coverage-preserving folding using shape indexing
4. Formal guarantees: termination, soundness, and completeness

Future Work

- Strategy language for customizable search heuristics
- Constructive symmetry breaking (frontier pruning)
- Richer OCL constraint language for MDE

Questions?

PISTEP®