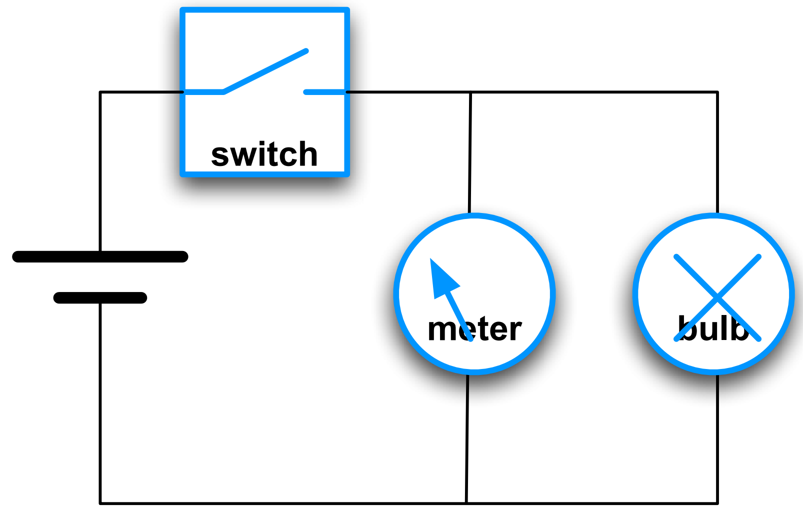
Symbolic Model Checking of Logics with Actions

Charles Pecheur `charles.pecheur@uclouvain.be`

Franco Raimondi `f.raimondi@cs.ucl.ac.uk`

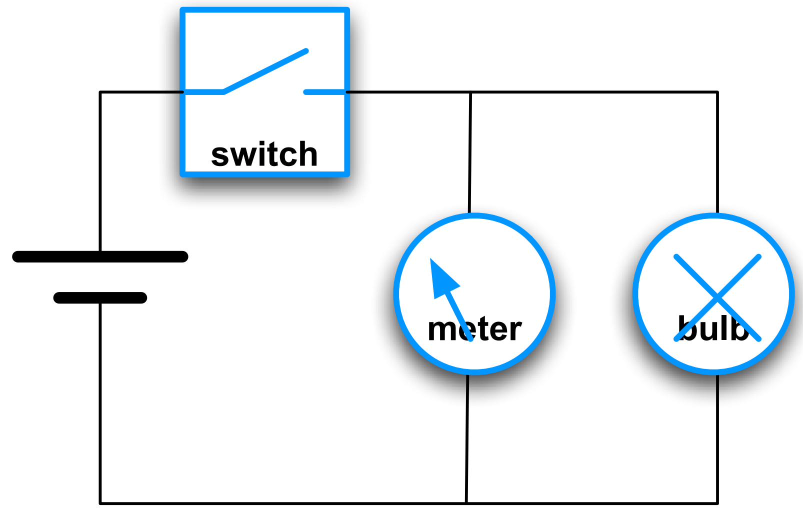
An Observable System

- An observable system
  - Electric circuit
- Hidden **state**
  - Bulb and meter can fail
- Visible **observations**
  - Read the meter, see the light



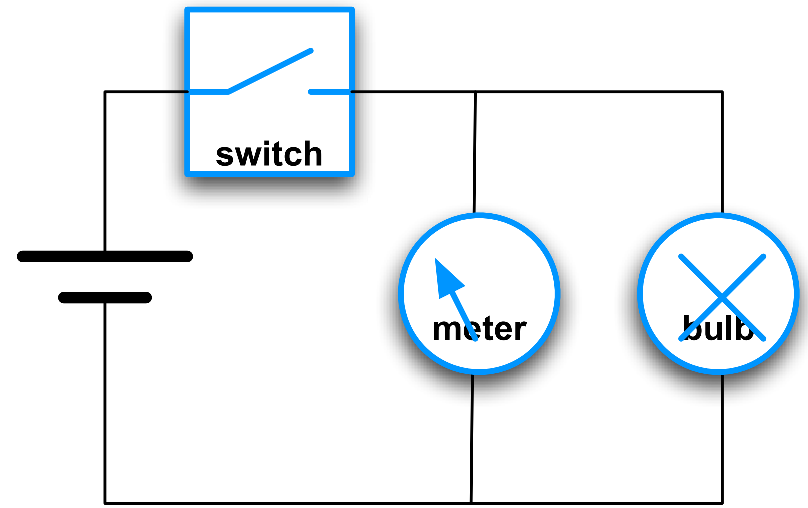
Diagnosis

- **Diagnosis**: from (history of) observations, infer state
  - Q: If there is no light and the meter reads zero, is there a current?
  - A: Maybe, if the meter is broken and the bulb has a short



Diagnosability

- **Diagnosability**:
  diagnosis is possible (up to desired precision, assuming context, . . .)
  - Q: Can I safely know when there is a short?
  - A: Yes, assuming single failures



From Diagnosability to Knowledge

- A condition F (fault) can be **diagnosed**

  iff

  an **agent** preceiving the observations (the diagnoser) always **knows** whether F holds or not
- In epistemic temporal logic CTLK:

$$\mathsf{AG}\left(\mathsf{K}_D\, F \vee \mathsf{K}_D\, \neg F\right)$$

where D is the diagnoser

From Knowledge to Actions

- In epistemic logic, agent A **knows** a fact ϕ

  iff

  ϕ is true in any possible state (world) consistent with A's knowledge
- Formalized as an **epistemic accessibility** (equivalence) relation $\sim_A$ between states that are indistinguishable by A
- We obtain a system with several transition relations $\langle \mathcal{S}, \mathcal{S}_0, \rightarrow, \sim_{A_1}, \ldots, \sim_{A_n} \rangle$ for n agents
- Or equivalently, a **labelled** transition system $\langle \mathcal{S}, \mathcal{S}_0, \mathcal{A}, \xrightarrow{a} \rangle$, where $\mathcal{A} = \{T, A_1, \ldots, A_n\}$

Model-Checking Diagnosability

- Custom diagnosability checker by Pecheur [MOCHART02,IJCAI03]
  - Uses NuSMV as back-end
  - **Idea: try epistemic approach instead?**
- Custom CTLK checker by Raimondi [TACAS06]
  - BDD-based, directly on interpreted systems
  - Very rudimentary modelling language
  - **Idea: use NuSMV instead?**
- Extend SMV to actions, then to CTLK
  - **This talk**

Outline

- Mixing states and actions: ARCTL
- Model checking of ARCTL
- ARCTL in SMV (two takes)
- Application to CTLK and experiments
- Related work, summary, perspectives

State-Based Temporal Logic

- The “classical” temporal logic
- Interpreted over executions of state machines, (unlabelled) transition systems
  - Atoms $\mathcal{P}_S$ are interpreted on **states**
  - **Kripke structure** (KS) $\langle \mathcal{S}, \mathcal{S}_0, \mathcal{R}, \mathcal{V} \rangle$, where $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S}$ and $\mathcal{V} : \mathcal{S} \rightarrow 2^{\mathcal{P}_S}$
- LTL, **CTL**, CTL*, μ-calculus, etc.
- Example: AG $(request \Rightarrow$ AF $response)$

Action-Based Temporal Logic

- Variant from the process algebra world
- Interpreted on **labelled** transitions systems
  - Atoms are **actions** (i.e. transition labels)
  - **labelled transitions system** (LTS) $\langle \mathcal{S}, \mathcal{S}_0, \mathcal{A}, \mathcal{T} \rangle$, where $\mathcal{T} \subseteq \mathcal{S} \times \mathcal{A} \times \mathcal{S}$
- No atoms on states; states are "not visible" (behavioural view)
- **Action-CTL** (ACTL) [deNicola-Vaandrager], ACTL*, Hennessy-Milner, etc.
- Example: $\mathsf{AG}_{true}\,(\neg \mathsf{EX}_{request}\,\mathsf{EG}_{\neg response}\,true)$

Mixing States and Actions

- Three generalizations:
  - Allow arbitrary atoms $\mathcal{P}_A$ interpreted over A
  - Allow both state and action atoms
  - Allow finite full-paths (i.e. sink states)
- **Mixed transition systems** (MTS) $\langle \mathcal{S}, \mathcal{S}_0, \mathcal{A}, \mathcal{T}, \mathcal{V}_S, \mathcal{V}_A \rangle$, where
  - $\mathcal{T} \subseteq \mathcal{S} \times \mathcal{A} \times \mathcal{S}$
  - $\mathcal{V}_S : \mathcal{S} \rightarrow 2^{\mathcal{P}_S}$
  - $\mathcal{V}_A : \mathcal{A} \rightarrow 2^{\mathcal{P}_A}$
- Contains LTS and KS as sub-structures

Action-Restricted CTL

- To support CTLK, we want to combine different transition/accessibility relations $\rightarrow$, $\sim_{A_1}, \ldots, \sim_{A_n}$ into a single labelled transition relation over alphabet $\mathcal{A} = \{T, A_1, \ldots, A_n\}$
- Correspondingly, we want to extend CTL so that temporal operators can be **restricted** to a given set of (or condition on) actions
  - e.g. over all T-paths, ϕ holds globally
- **Action-Restricted CTL** (ARCTL) generalizes path quantifiers A, E into A_α, E_α restricted to α-paths

ARCTL Semantics

- Given a mixed transition system $\mathcal{M}$, let
  - $\Pi(s)$ the set of (finite or infinite) full-paths of $\mathcal{M}$ from s
  - α a propositional formula over $\mathcal{P}_A$
  - $\mathcal{M}|_\alpha$ the restriction of $\mathcal{M}$ to α-actions
  - $\Pi|_\alpha(s)$ the set of full-paths of $\mathcal{M}|_\alpha$ from s
- For a path formula γ, we have
  - $s \models \mathsf{A}_\alpha\gamma$ iff $\forall\pi \in \Pi|_\alpha(s) \cdot \pi \models \gamma$
  - $s \models \mathsf{E}_\alpha\gamma$ iff $\exists\pi \in \Pi|_\alpha(s) \cdot \pi \models \gamma$

  (full formal definitions in the paper)

ARCTL Properties and Remarks

- Obviously $\mathsf{E}_{true}\gamma \equiv \mathsf{E}\gamma$ and $\mathsf{A}_{true}\gamma \equiv \mathsf{A}\gamma$
- In general $\Pi|_\alpha(s) \not\subseteq \Pi(s)$ and thus $\mathsf{E}_\alpha\gamma \not\Rightarrow \mathsf{E}\gamma$
  - because a (finite) α-full-path may only be a prefix of a (finite or infinite) full-path
  - e.g. $\mathsf{E}_a\mathsf{G}\, p \not\Rightarrow \mathsf{EG}\, p$ on $p \xrightarrow{a} p \xrightarrow{b} \bar{p}$
- In comparison, ACTL puts action conditions on temporal quantifiers,
  - e.g. $\mathsf{EF}_\alpha\, \phi$ = "**all** paths **are** α-paths until they reach ϕ"
  - Not adequate for our purpose

Finite Paths

Unlike classical definitions of model-checking, we allow **finite full-paths**

- Even with infinite full-paths, we would have finite α-full-paths anyway
- The semantics of CTL (and thus ARCTL) generalizes nicely
  - This is not new
- In particular, $\pi \models \mathsf{X}\,\phi$ iff $\underline{|\pi| \geq 1} \wedge \pi(1) \models \phi$
  - $\mathsf{E}_\alpha \mathsf{X}\,\phi$ xor $\mathsf{A}_\alpha \mathsf{X}\,\neg\phi$ xor $\neg \mathsf{E}_\alpha \mathsf{X}\,true$
- $\mathsf{G}\,\phi$ holds for finite α-full-paths where ϕ holds
  - We define $\mathsf{G}^\omega\,\phi$ for infinite paths only

Model Checking of ARCTL

Generalizes CTL model checking:

- All ARCTL operators can be reduced to $\mathsf{E}_\alpha\mathsf{X}$ and $\mathsf{E}_\alpha\mathsf{U}$ and $\mathsf{E}_\alpha\mathsf{G}^\omega$
  - Additional conditions w.r.t. finite paths
  - e.g. $\mathsf{A}_\alpha\mathsf{F}\,\phi \equiv \neg\mathsf{E}_\alpha[\neg\phi\,\mathsf{U}\,\neg\phi \wedge \neg\mathsf{E}_\alpha\mathsf{X}\,true] \wedge \neg\mathsf{E}_\alpha\mathsf{G}^\omega\,\neg\phi$
- Given sets of states S, S' and actions $A \in 2^{\mathcal{A}}$, we define functions $eax(A,S)$, $eau(A,S,S')$ and $eag(A,S)$ capturing the semantics of those operators
  - e.g. $eau(A,S,S') = \mu Z \cdot S' \cup (S \cap eax(A,Z))$
- For any formula ϕ we can compute $[\![\phi]\!]$ using these functions
- This can all be computed using BDDs

Model Checking: details

$$
\begin{aligned}
eax(A,S) &= \{s \mid \exists a, s' \cdot s \xrightarrow{a} s' \land a \in A \land s' \in S\} \\
eau(A,S,S') &= \mu Z \cdot S' \cup (S \cap eax(A,Z)) \\
eag(A,S) &= \nu Z \cdot S \cap eax(A,Z)
\end{aligned}
$$

$$
\begin{aligned}
[\![\mathsf{E}_\alpha \mathsf{X}\, \phi]\!] &= eax([\![\alpha]\!], [\![\phi]\!]) \\
[\![\mathsf{A}_\alpha \mathsf{X}\, \phi]\!] &= \underline{eax([\![\alpha]\!])} \cap \neg eax([\![\alpha]\!], \neg[\![\phi]\!]) \\
[\![\mathsf{E}_\alpha (\phi\, \mathsf{U}\, \phi')]\!] &= eau([\![\alpha]\!], [\![\phi]\!], [\![\phi']\!]) \\
[\![\mathsf{A}_\alpha (\phi\, \mathsf{U}\, \phi')]\!] &= \neg eau([\![\alpha]\!], \neg[\![\phi']\!], \neg[\![\phi']\!] \cap (\neg[\![\phi]\!] \cup \underline{\neg eax([\![\alpha]\!])})) \\
&\quad \cap \neg eag([\![\alpha]\!], \neg[\![\phi']\!])
\end{aligned}
$$

Finite Paths and Fairness

- CTL can be verified modulo **fairness conditions**
  - Sets of sets of states that fair traces visit infinitely often
  - LTL is reducible to CTL+fairness
- Could be extended to labelled paths and ARCTL
  - With fairness conditions on states and actions
- However, by definition, **finite full-paths are unfair**
  - A revised notion of fairness (with model checking solution) is needed
  - For further investigation . . .

SMV (with Actions)

- NuSMV: symbolic model checker (IRST)
  - Rich modular modeling language
  - Properties in CTL
  - Many features, open-source
- The SMV language supports actions!
  - Named **input variables** (IVARs)
  - Unfortunately, may appear only in the **model**, **not** in the CTL **properties**

ARCTL in SMV (Take One)

First approach:

- Reduce **mixed transition structure** $\mathcal{M}$ to **Kripke structure** $post(\mathcal{M})$
- Reduce **ARCTL formula** ϕ to plain **CTL formula** $post(\phi)$

Such that

$$(\mathcal{M}, s) \models \phi \quad \text{iff} \quad (post(\mathcal{M}), s) \models post(\phi)$$

- Check $(post(\mathcal{M}), s) \models post(\phi)$ in NuSMV
  - Does not use IVARs

Post-Projection of Actions

Principle:

- Project action propositions into the next state
  - $\mathcal{S}' = \mathcal{A} \times \mathcal{S}$, $\mathcal{P}' = \mathcal{P}_S \cup \mathcal{P}_A$
  - $s \xrightarrow{a} s'$ becomes $(*, s) \longrightarrow (a, s')$, for any action $*$
- Reduce ARCTL to CTL accordingly, e.g.

$$
\begin{aligned}
post(\mathsf{E}_\alpha \mathsf{X}\, \phi) &= \mathsf{EX}\,(\alpha \wedge post(\phi)) \\
post(\mathsf{A}_\alpha \mathsf{X}\, \phi) &= \mathsf{AX}\,(\alpha \Rightarrow post(\phi)) \wedge \underline{\mathsf{EX}\,\alpha} \\
post(\mathsf{A}_\alpha(\phi\, \mathsf{U}\, \phi')) &= post(\phi') \vee (\underline{\mathsf{EX}\,\alpha} \wedge post(\phi) \\
&\quad \wedge \mathsf{AX}\,\mathsf{A}[\underline{\mathsf{EX}\,\alpha} \wedge post(\phi)\, \mathsf{U}\, \neg\alpha \vee post(\phi')])
\end{aligned}
$$

  - **Erratum:** underlined terms missing in paper

Post-Projection in SMV

Both reductions have been implemented as M4 macros

- `TRANS_A(a,t)` $\mapsto$
  `TRANS next(a) -> (t)`
- `EU_A(a,p,q)` $\mapsto$
  `(((p) & EX E[(a) & (p) U (a) & (q)]) | (q))`
  - where a is an action formula, p, q are state formulae, `t` is a transition constraint
- User has to decide which variables are for actions (a) and which are for states (p, q)

ARCTL in SMV (Take Two)

Second approach: extend NuSMV to provide native support for ARCTL

- Use IVARs for action variables
  - Any **valuation** of IVARs is a different action label
- Extended syntax EAX (α) ϕ, EA (α) [ϕ U ϕ'], etc.
- Implementation of $eax(A, S)$, $eau(A, S, S')$ and $eag(A, S)$ on BDDs
  - As variants of existing $ex(S)$, $eu(S, S')$ and $eg(S)$
- Not done yet: generation of counter-examples

CTLK in ARCTL

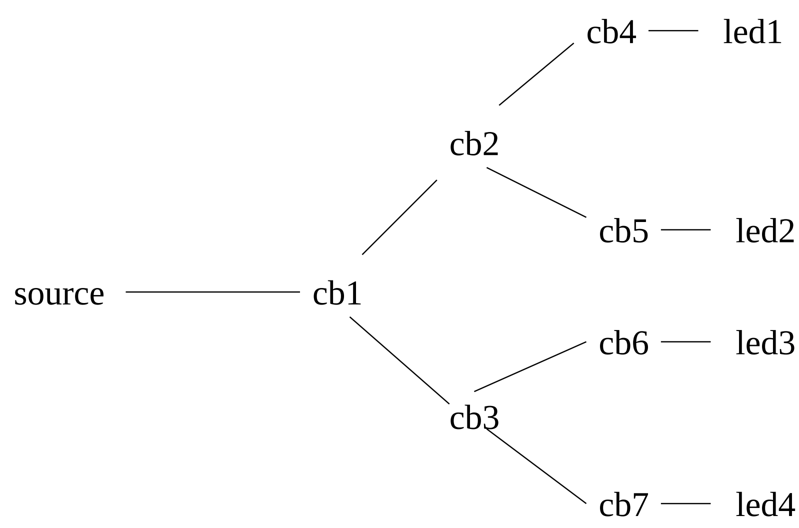
Principle: temporal transitions $s \rightarrow s'$ and epistemic accessibility relations $s \sim_{A_i} s'$ become different labels of a single labelled transition relation

- Multi-agent system (MAS) model $\mathcal{M}_K$ translated to MTS model $F(\mathcal{M}_K)$
- CTLK property ϕ_K translated to corresponding ARCTL property $F(\phi_K)$
  - e.g. $F(\mathsf{K}_A\ \phi) = \mathsf{A}_A\mathsf{X}\,(reachable \Rightarrow F(\phi))$
- Both translations implemented as M4 macros
- Model checked in SMV using either the native extension to ARCTL or further reduction to plain CTL
- Details in forthcoming paper…

Experiments 1

First experiment: verify diagnosability expressed in CTLK on *circuit-breaker* example

- Example from Livingstone model-based diagnosis system
  - cascade of circuit breakers
  - Automatically translated to SMV



Results 1

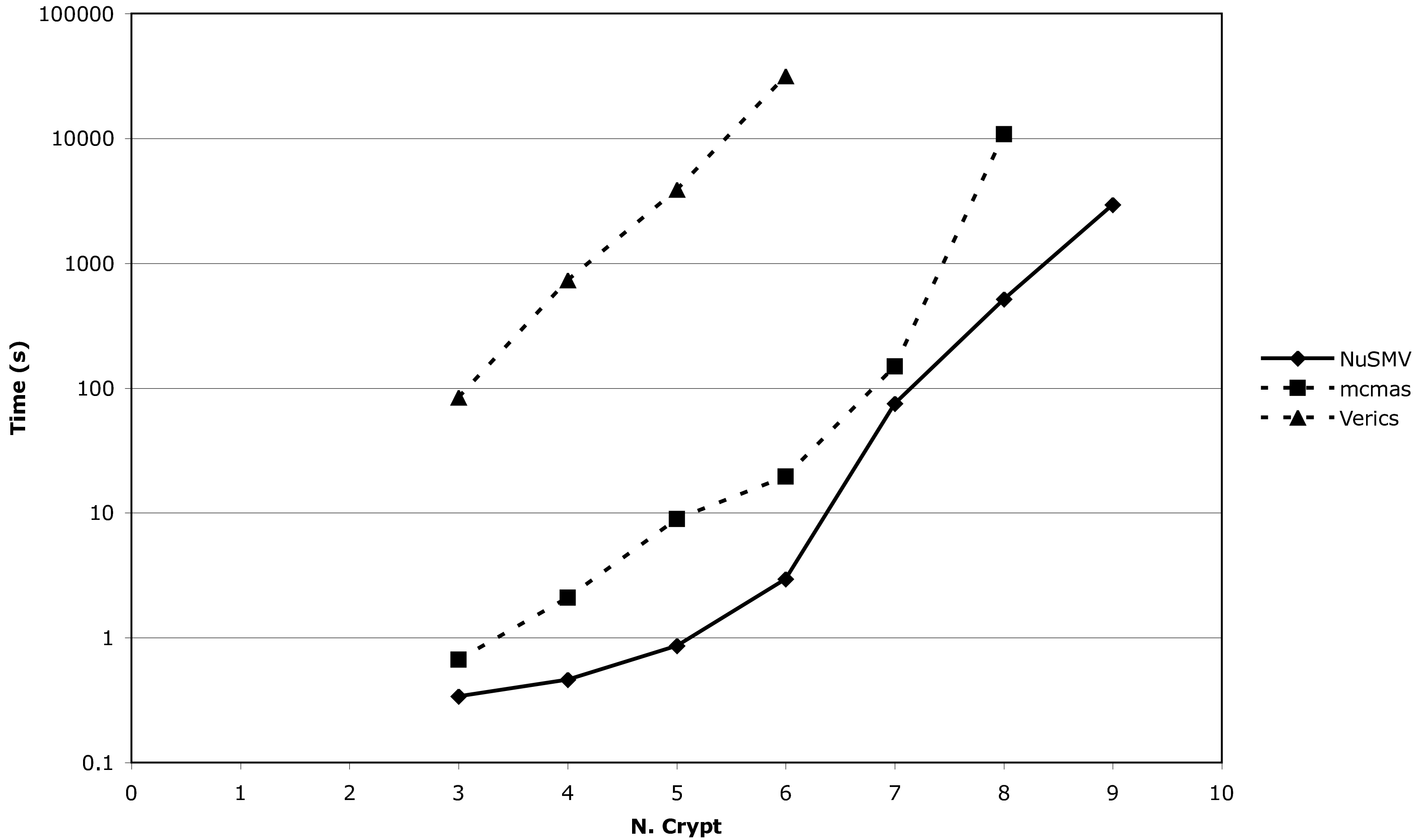
- Diagnosability property: $\mathsf{AG}\,(\mathsf{K}_D\,(faulty) \vee \mathsf{K}_D\,(\neg faulty))$
- Used native ARCTL implementation
- Tried for various model sizes (depth of the cascade)
- Verified up to 240-bit states in less than 10 min
  - Performance similar to factory NuSMV on plain CTL properties

Experiments 2

Second experiment: verify CTLK properties of the *Dining Cryptographers protocol*

- Not diagnosis, Classical example for general epistemic properties
  - Scalable number N of agents (the Cryptographers)
- Verified protocol correctness properties
- Results are not in this paper, submitted
- 99-bit state for $N = 5$
- Comparison with Verics [Penczek et al.], MCMAS [Raimondi et al.]

Results 2



100000
10000
1000
100
10
1
0.1
Time (s)
0
1
2
3
4
5
6
7
8
9
10
N. Crypt
NuSMV
mcmas
Verics

Related Work

- Other action-based logic model checkers:
  - EST [Meolic et al.] for variant of ACTL
  - SAM [Fancheti et al.] for ACTL with fixpoint operators

  No state conditions, no SMV language for modeling
- Encoding of process algebras as BDDs by [Enders et al., Dsouza et al.]
- Reduction from ACTL to CTL by [de Nicola and Vaandrager]
  - In original ACTL paper
  - Adds intermediate state for every transition

Summary

- Main contributions:
  - **ARCTL**, a branching temporal logic with **action-based and state-based** atoms
  - A **reduction** $post$ from ARCTL to CTL (with corresponding reduction on models)
  - A generalization of **BDD-based model-checking** from CTL to ARCTL
  - **Two implementations** of ARCTL **in SMV**: native and using $post$
- Context: **diagnosability** reduces to **CTLK**, which reduces to ARCTL
- Early but promising **experimental results**

Perspectives

- Further work:
  - Add generation of **counter-examples**
  - Study **weak variants** of ARCTL (i.e. ignoring internal actions)
  - Handle **fairness**
- Possible extensions:
  - Use SAT-based **bounded model checking** (restricts supported formulae)
  - Generalize to **game-theoretic** logics such as ATL