A Temporal Logic for Multi-threaded Programs

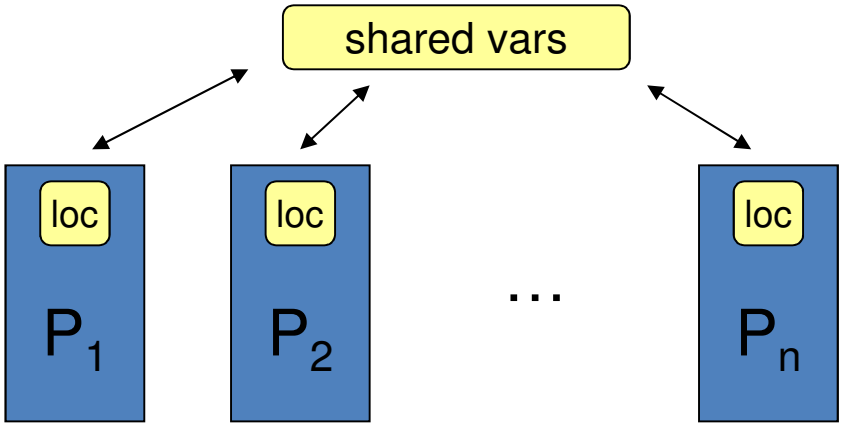
Salvatore La Torre Margherita Napoli

Dipartimento di Informatica
Università degli Studi di Salerno

TCS 2012

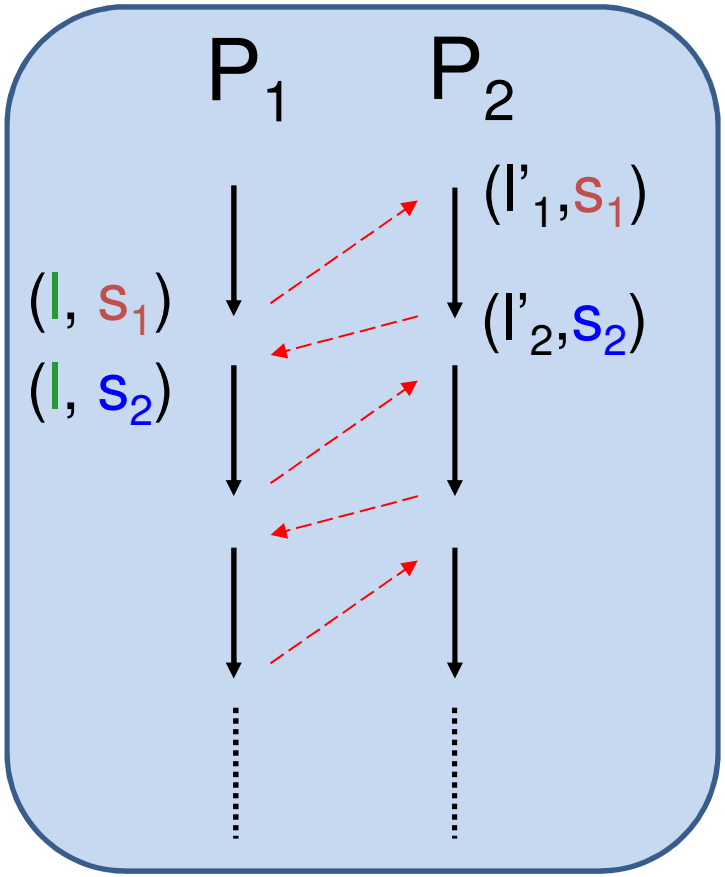
Multi-threaded program (MP)

- A fixed number of programs $P_1, \ldots, P_n$ (running in parallel)
  - each program has (recursive) procedure calls
- communication is through shared variables

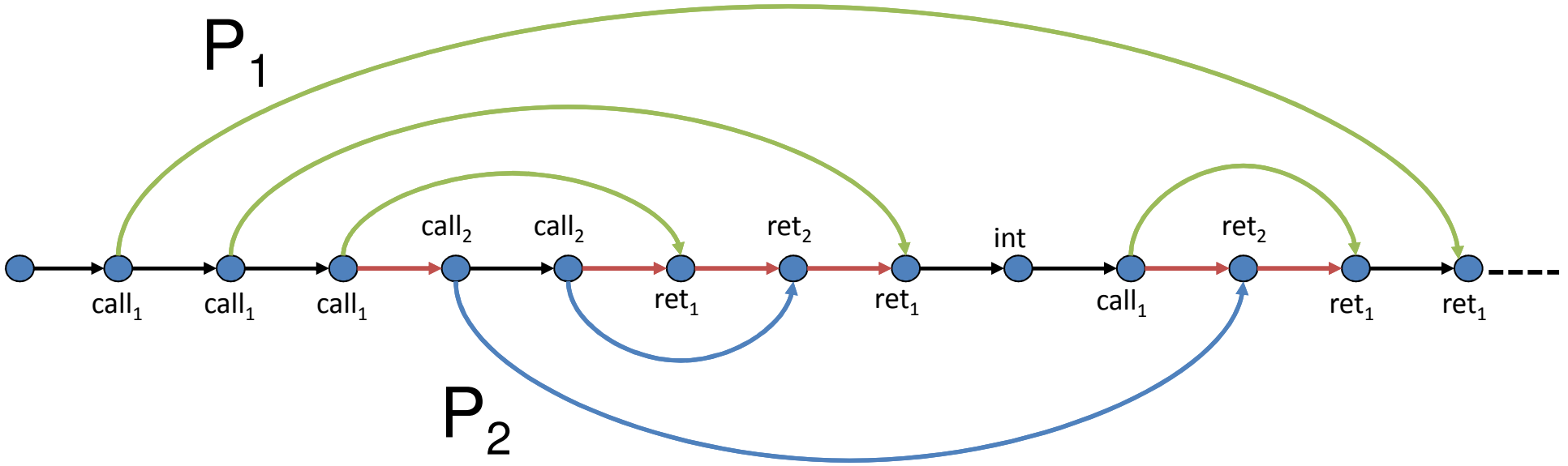


Computation of MP

- Global states (l,g)
  - g is a valuation of shared variables (*shared state*)
  - l is *local state*
- Semantics by interleaving:
  - computations as sequences of execution contexts
  - only one P_i is active in each context
  - either the active P_i moves (*local behavior*)
  - or control switches to another component (*context-switch*)



Computation as multiply nested word



- matching calls and returns are connected by curved edges (green for P_1 and blue for P_2)
  - each set of edges defines a matching relation
- linear edges capture the linear ordering in the interleaved sequence (red edges denote the context switches)

Our focus

- A temporal logic to specify properties of multi-threaded programs
- LTL [Pnueli,FOCS77] is a popular choice for concurrent finite-state programs
- We allow recursive procedure calls (infinite state)
- We introduce the logic MultiCaRet that explicitly refers to call-returns within each thread

MultiCaRet

- extends to multi-threaded programs the logic CaRet [Alur/Etessami/Madhusudan, TACAS'04]
- typical linear time temporal logic operators with superscripts

$$\varphi := p \mid call_i \mid ret_i \mid \neg\varphi \mid \varphi \vee \varphi \mid \bigcirc \varphi \mid \varphi \, U \, \varphi \mid$$
$$\bigcirc^{a_i} \varphi \mid \varphi \, U^{a_i} \varphi \mid \bigcirc^{-i} \varphi \mid \varphi \, U^{-i} \varphi$$

  - " i " refers to the i^{th} thread
  - " a " refers to abstract successors (on calls jump directly to matching returns)
  - " - " refers to call successors (down into the call-stack)

Decision problems

- Satisfiability:
  - φ be a MultiCaRet formula

  Is there a computation that satisfies φ ?

- Model-checking:
  - φ be a MultiCaRet formula
  - P be a multi-threaded program

  Do all computations of P satisfy φ ?

Our contribution

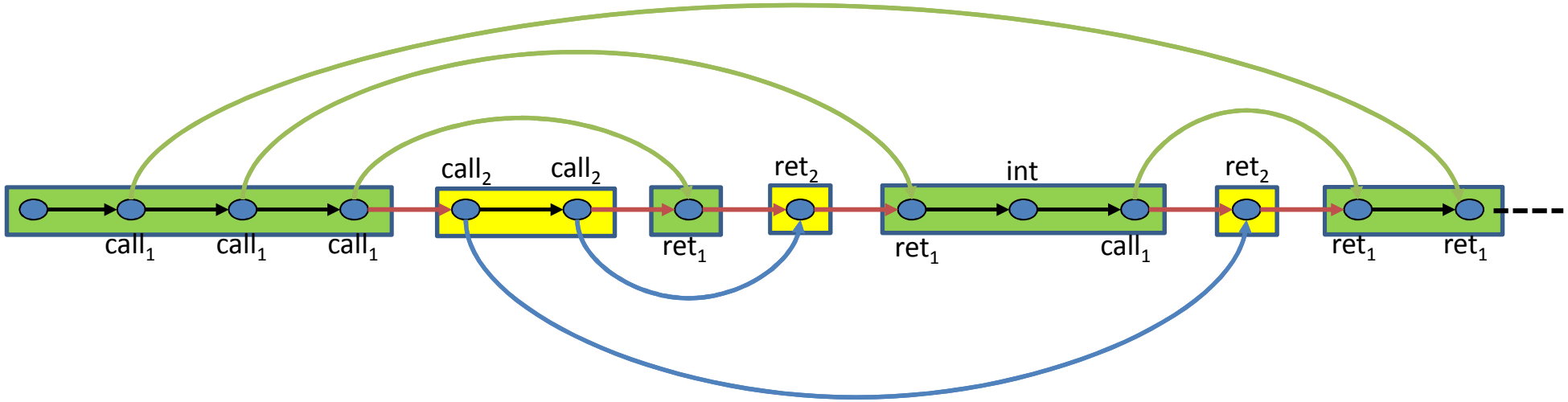
- Automata theoretic approach to solve such decision probls
- Decidability results by restricting the program behaviors
  - the considered problems are in general undecidable (two threads suffice to encode a Turing Machine)
- In particular,

MultiCaRet model-checking and satisfiability probls are EXPTIME-complete for scope-bounded computations

  - with this restriction reachability is PSPACE-complete [La Torre/Napoli, CONCUR'11]

k-scoped computations

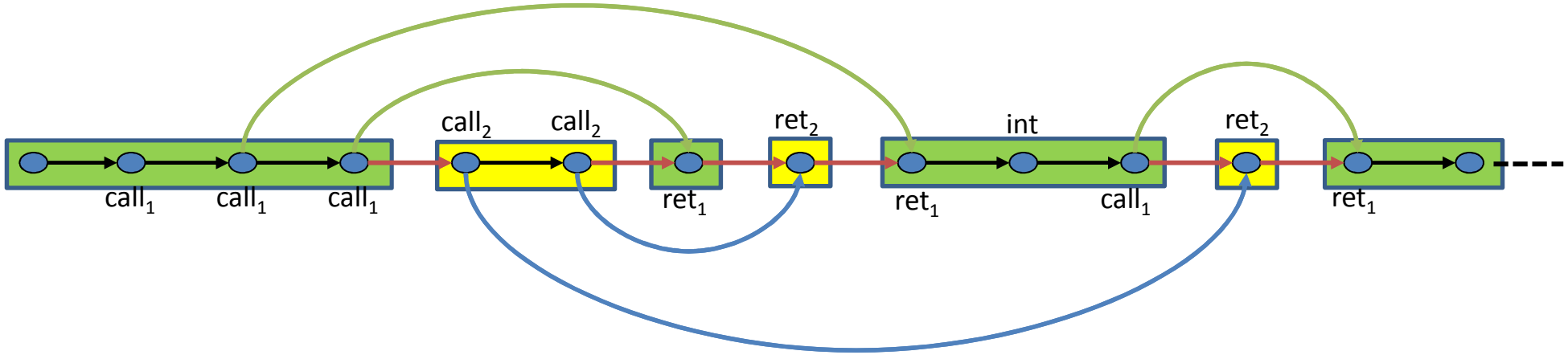
A call in thread "i" can be returned only within the next k contexts of this thread



– This portion is 4-scoped

k-scoped computations

A call in thread "i" can be returned only within the next k contexts of this thread



– This portion is 3-scoped

Automata-theoretic approach

- construct a Büchi multistack pushdown system M_φ capturing all models of formula φ
  - satisfiability reduces to checking "$L(M_\varphi) \neq \emptyset$"
  - model-checking reduces to checking

    "$L(M_P) \cap L(M_\varphi) = \emptyset$"

- construction of M_φ extends tableau-based constructions for temporal logic and

  $|M_\varphi| = 2^{O(|\varphi|)}$

Emptiness of Büchi n-stack MPS

- undecidable in general

- Given an n-stack k-scoped MPS M we show

Theorem.

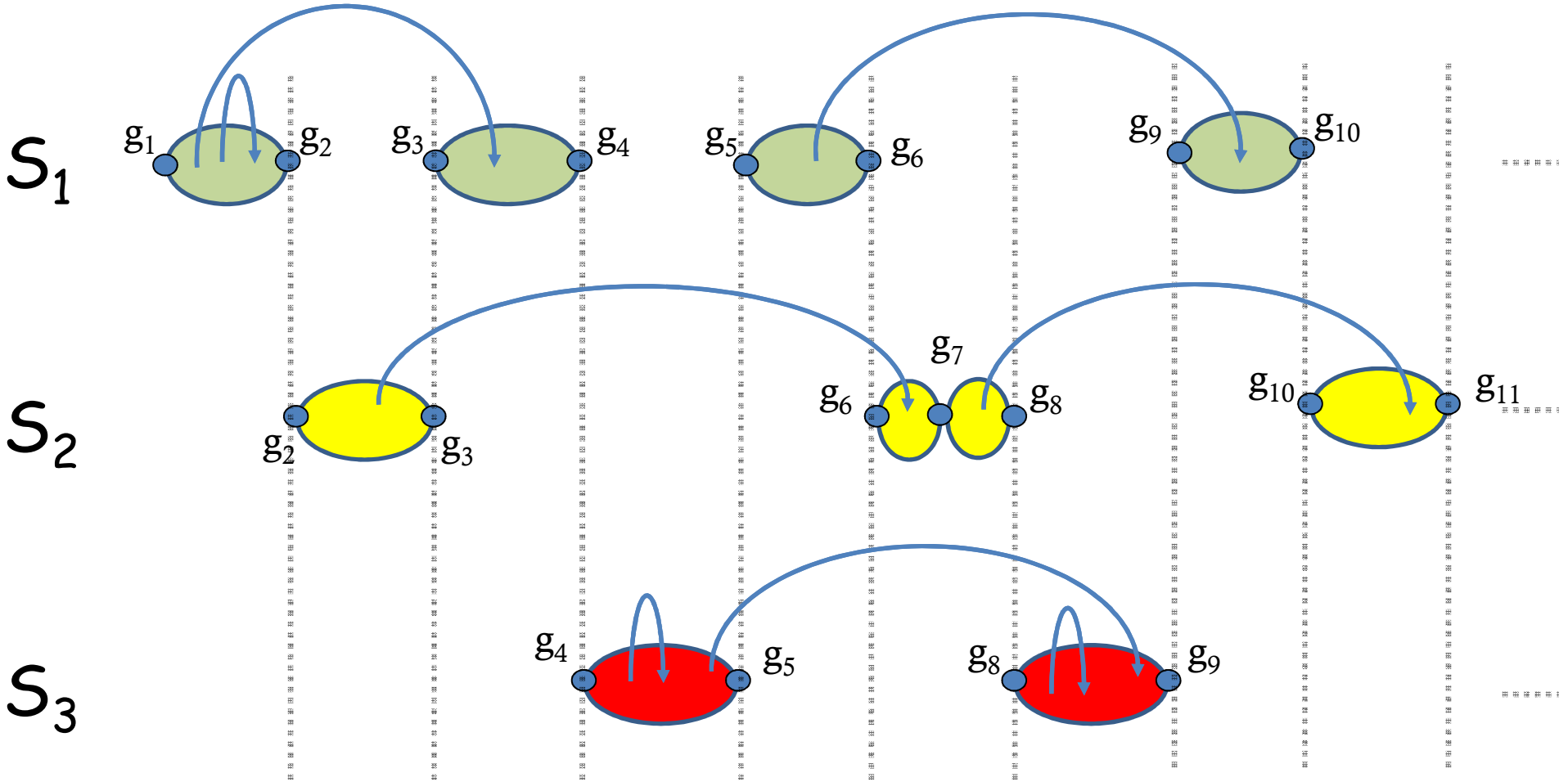
There exists a Büchi automaton A_M of size $|M|^{O(nk)}$ such that

$$L(M)=\emptyset \text{ iff } L(A_M)=\emptyset$$

Rest of the talk

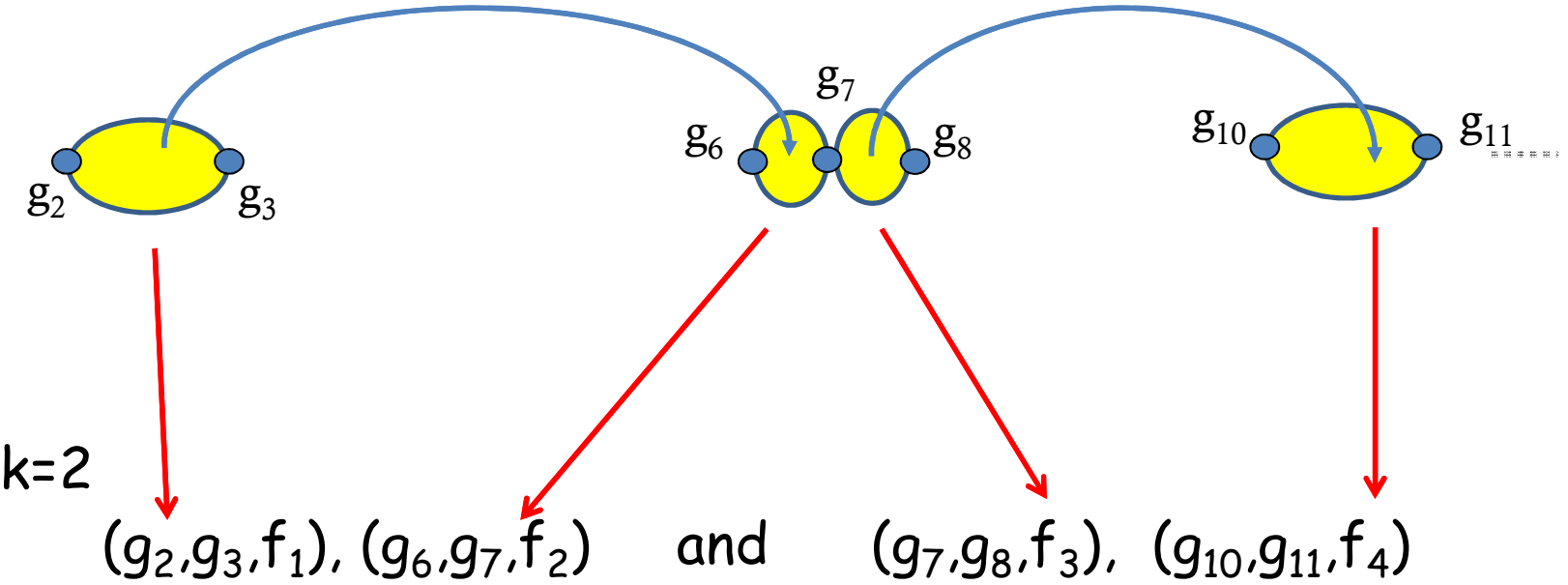
- Construction of the Büchi automaton A_M
- Proof of the Theorem
- Conclusion and related work

A 2-scoped computation of a 3-stack MPS



Construction of the Büchi automaton

Collect thread-interfaces of length ≤k

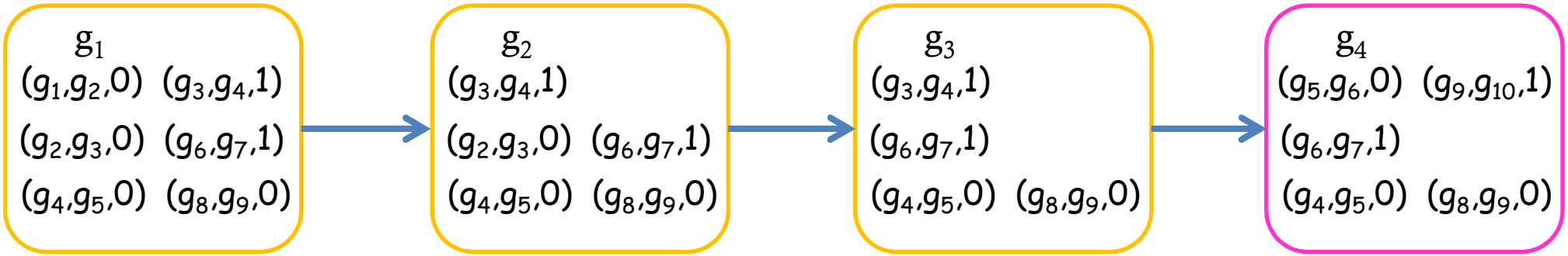


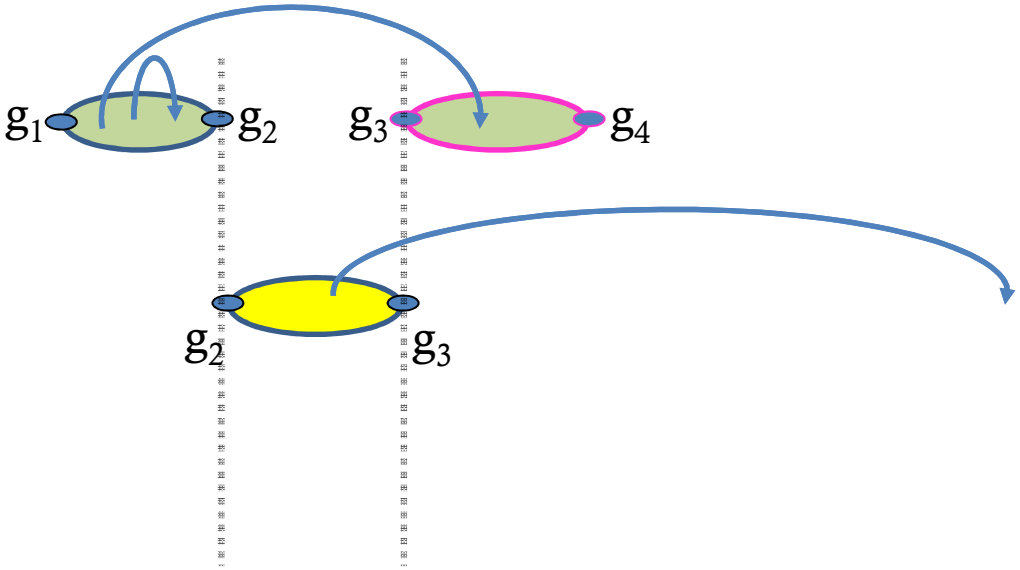
f-component is true iff a final state is visited in the corresponding portion of run

(keeps track of the Büchi acceptance)

Construction of the Büchi automaton

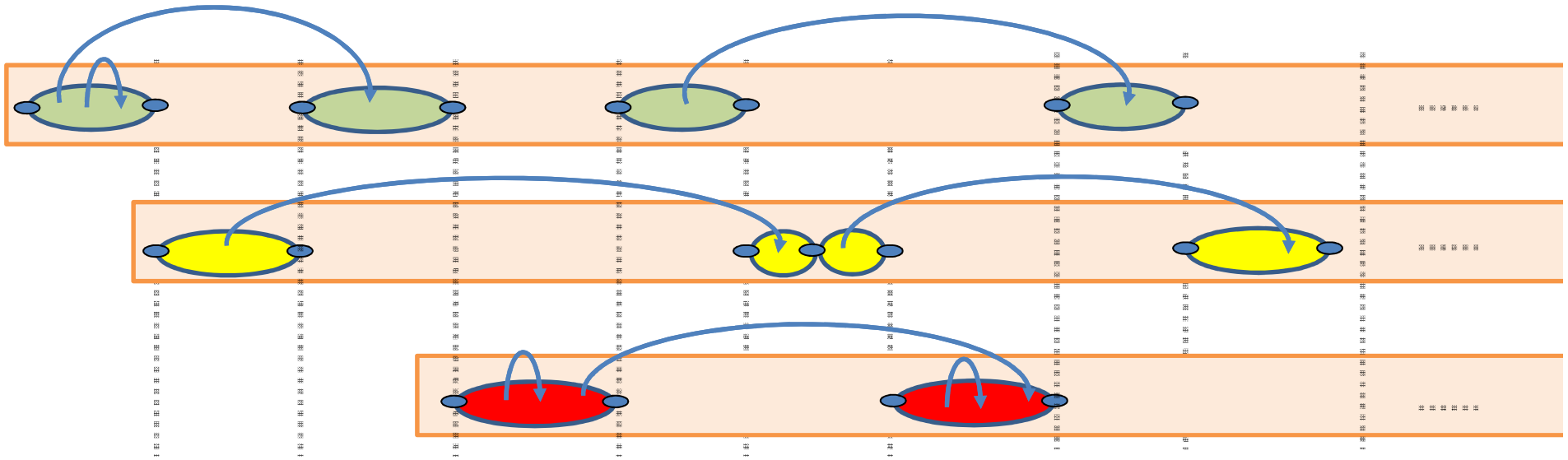
Mimic the MPS behavior via the thread-interfaces





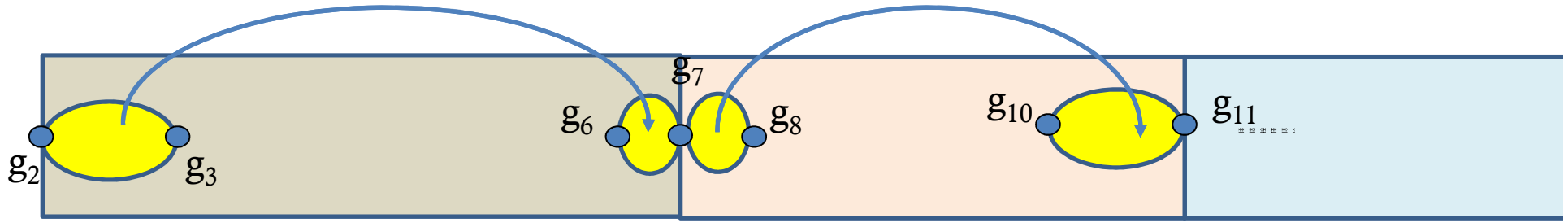
Correctness

- ω-thread-interfaces (ω-TI) suffice to capture all runs of an MPS
  - each run corresponds to the composition of n ω-TI



Correctness

- for k-scoped runs, each such ω-TI is the concatenation of thread-interfaces of length ≤k



- Recall: a call can be returned only within the next k contexts of this thread
- After k contexts we can continue as starting from empty stack

Complexity

- the states of the Büchi automaton A_M are
$$(q, X_1,\ldots,X_n, f)$$
  - $f \in \{0,1\}$ signals when a final state is visited
  - each X_i is a sequence of at most k tuples from
$$Q_M \times Q_M \times \{0,1\}$$

- size of A_M is $|M|^{O(nk)}$

Rest of the talk

- ✓ Construction of the Büchi automaton A_M
- ✓ Proof of the Theorem
- Conclusion and related work

More results

- "ordered" MPS (OMPS)
  [Breveglieri/Cherubini/Citrini/Crespi-Reghizzi,IJFOCS'96]

- Emptiness of Büchi OMPS is 2ETIME-complete
  - lower bound in [Atig/Bollig/Habermehl,DLT'08]
  - upper bound in [Atig,FSTTCS'10]

- By automata theoretic approach both MultiCaRet model checking and satisfiability 2ETIME-complete for "ordered" computations

More results

- We give a FO characterization of MultiCaRet
  - Decidability of MultiCaRet can be extended to any MSO-definable class of multiply nested words of bounded tree-width

    (use [Madhusudan/Parlato, POPL'11])

- Our notion of scope-bounded computations is less restrictive than in [La Torre/Napoli, CONCUR'11]
  - Now unboundedly many contexts may happen between two consecutive contexts of a given thread

Choice of TL

- MultiCaRet looks suitable to express TL properties in multi-threaded programs

- Focusing on multi-nested words, the natural TL is an extension of NWTL [Alur et al., LMCS'08]
  - tableau construction still feasible
  - our decidability results still hold

Bounded context/phase switching

- reachabiltiy is decidable also when restricting to
  - computations up to a bounded number of context-switching [Qadeer/Rehof,TACAS'05]
  - computations up to a bounded number of phase-switching [La Torre/Madhusudan/Parlato, LICS'07]

- bounded context and phase switching
  - do not "scale" to ω-computations (ω-suffix is all in 1 context/phase)
  - covered by the considered restrictions

More related work

- A general TL subsuming MultiCaRet is introduced in [Bollig/Cyriac/Gastin/Zeitoun, MFCS'11]
  - decidability given w.r.t. phase-bounded computations
- LTL model-checking is addressed for
  - scope-bounded computations (via scope-bounded Büchi MPS) in [Atig/Bouajjani/Kumar/Saivasan, ATVA'12]
  - "ordered" computations in [Atig, FSTTCS'10]
- Concurrent ext. of CaRet [Bozzelli/La Torre/Peron, TCS'08]
- Sequentialization and bounded treewidth for scope-bounded computations [La Torre/Parlato, FSTTCS'12]