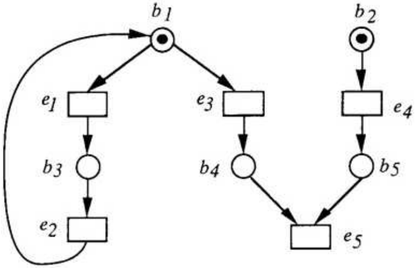
Model-Checking Event Structures, Part 2

Madhavan Mukund

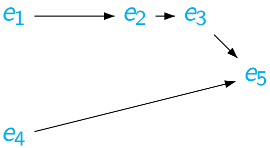
Chennai Mathematical Institute
http://www.cmi.ac.in/˜madhavan

Formal Methods Update Meeting
IIT Roorkee
14 July 2009

Concurrent systems

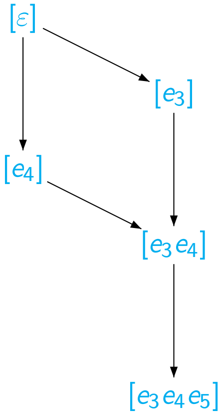


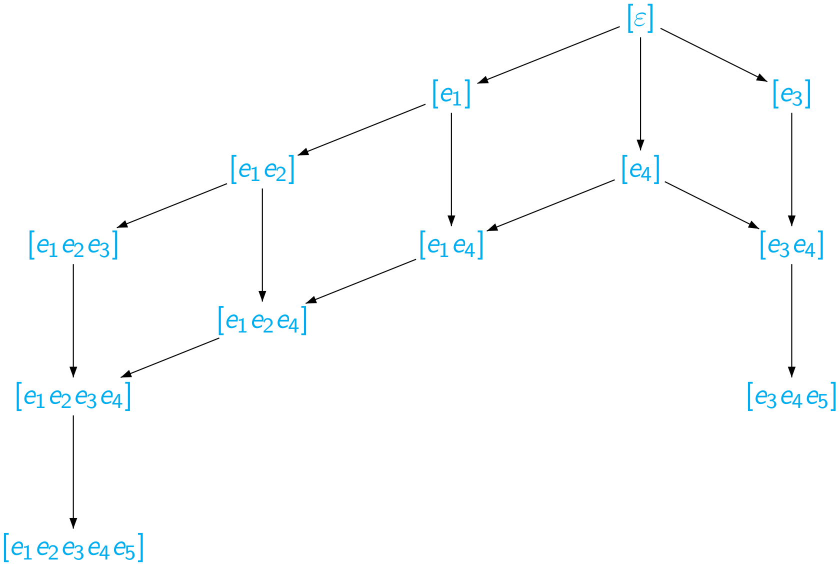
- Convenient to view each execution as a labelled partial order



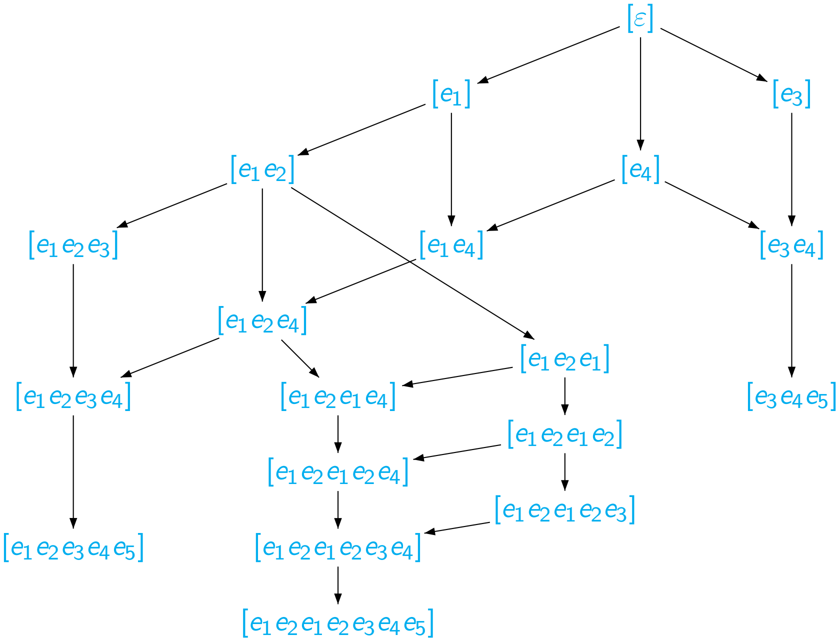
Mazurkiewicz traces

- Actions are enriched with independence relation specifying which pairs are independent
  - Symmetric, irreflexive
  - Typically derived from structure of underlying system
    - Actions performed by disjoint sets of components
- In a linearization, adjacent independent actions can be swapped to yield an equivalent linearization


[ε]
[e3]
[e4]
[e3 e4]
[e3 e4 e5]



[ε]
[e1]
[e3]
[e1 e2]
[e4]
[e1 e2 e3]
[e1 e4]
[e3 e4]
[e1 e2 e4]
[e1 e2 e3 e4]
[e3 e4 e5]
[e1 e2 e3 e4 e5]


[ε]
[e1]
[e3]
[e1 e2]
[e4]
[e1 e2 e3]
[e1 e4]
[e3 e4]
[e1 e2 e4]
[e1 e2 e1]
[e1 e2 e3 e4]
[e1 e2 e1 e4]
[e3 e4 e5]
[e1 e2 e1 e2]
[e1 e2 e1 e2 e4]
[e1 e2 e1 e2 e3]
[e1 e2 e3 e4 e5]
[e1 e2 e1 e2 e3 e4]
[e1 e2 e1 e2 e3 e4 e5]

From traces to event structures

- Can extract an event structure from the set of traces

From traces to event structures

- Can extract an event structure from the set of traces
- $t \leq t'$ if t' extends t with more events
  - For instance, $[e_1 e_2 e_3] \leq [e_1 e_4 e_2 e_3]$

From traces to event structures

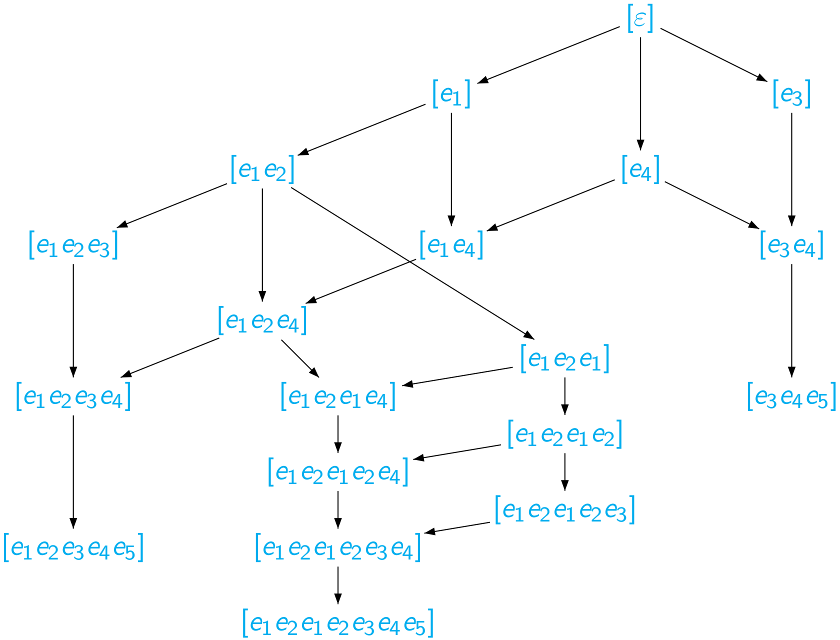
- Can extract an event structure from the set of traces
- $t \leq t'$ if t' extends t with more events
  - For instance, $[e_1 e_2 e_3] \leq [e_1 e_4 e_2 e_3]$
- t and t' are compatible if there is t'' such that $t \leq t''$ and $t' \leq t''$
  - For instance, $[e_1 e_2 e_3]$ and $[e_4]$ are compatible because both are dominated by $[e_1 e_2 e_3 e_4]$

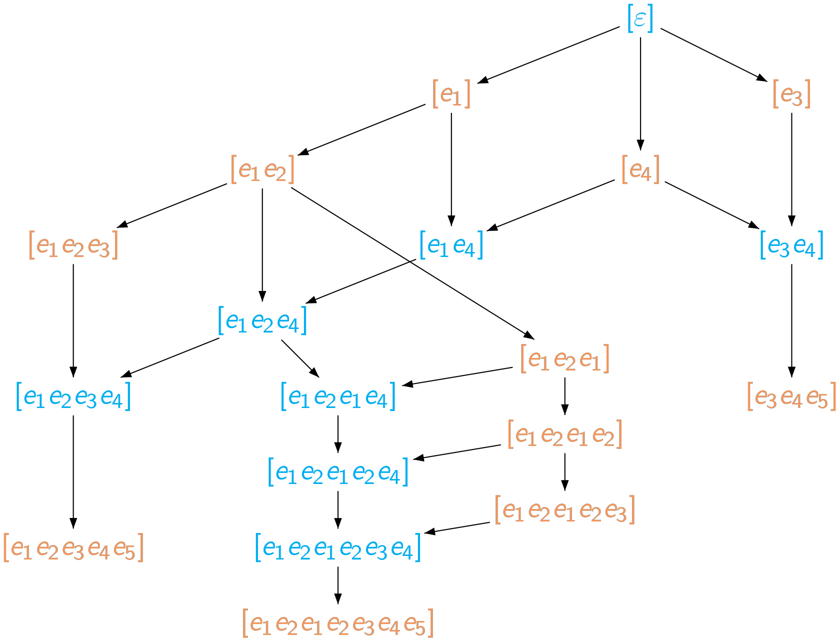
From traces to event structures

- Can extract an event structure from the set of traces
- $t \leq t'$ if t' extends t with more events
  - For instance, $[e_1 e_2 e_3] \leq [e_1 e_4 e_2 e_3]$
- t and t' are compatible if there is t'' such that $t \leq t''$ and $t' \leq t''$
  - For instance, $[e_1 e_2 e_3]$ and $[e_4]$ are compatible because both are dominated by $[e_1 e_2 e_3 e_4]$
- $t \# t'$ if t and t' are not compatible

From traces to event structures

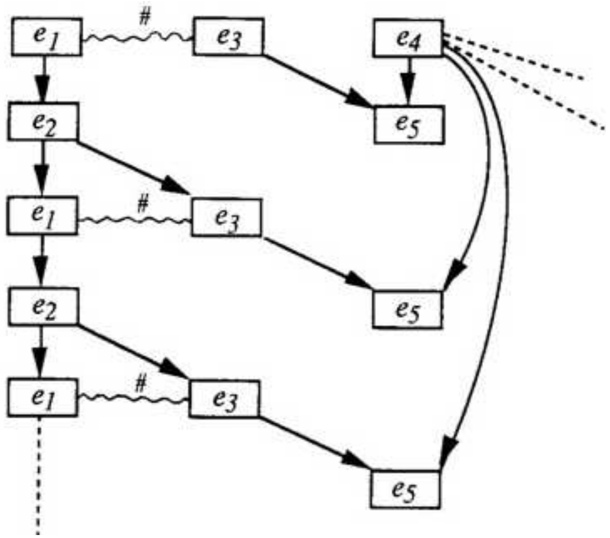
- Can extract an event structure from the set of traces
- $t \leq t'$ if t' extends t with more events
  - For instance, $[e_1 e_2 e_3] \leq [e_1 e_4 e_2 e_3]$
- t and t' are compatible if there is t'' such that $t \leq t''$ and $t' \leq t''$
  - For instance, $[e_1 e_2 e_3]$ and $[e_4]$ are compatible because both are dominated by $[e_1 e_2 e_3 e_4]$
- $t \# t'$ if t and t' are not compatible
- Identify events with prime traces
  - Prime trace: Only one maximal element
  - “Earliest” occurrence of an action


[ε]
[e1]
[e3]
[e1 e2]
[e4]
[e1 e2 e3]
[e1 e4]
[e3 e4]
[e1 e2 e4]
[e1 e2 e1]
[e1 e2 e3 e4]
[e1 e2 e1 e4]
[e3 e4 e5]
[e1 e2 e1 e2]
[e1 e2 e1 e2 e4]
[e1 e2 e1 e2 e3]
[e1 e2 e3 e4 e5]
[e1 e2 e1 e2 e3 e4]
[e1 e2 e1 e2 e3 e4 e5]



[ε]
[e1]
[e3]
[e1 e2]
[e4]
[e1 e2 e3]
[e1 e4]
[e3 e4]
[e1 e2 e4]
[e1 e2 e1]
[e1 e2 e3 e4]
[e1 e2 e1 e4]
[e3 e4 e5]
[e1 e2 e1 e2]
[e1 e2 e1 e2 e4]
[e1 e2 e1 e2 e3]
[e1 e2 e3 e4 e5]
[e1 e2 e1 e2 e3 e4]
[e1 e2 e1 e2 e3 e4 e5]

Event Structures ...



(Labelled) Event Structures

Formally, an event structure is of the form $ES = (E, \leq, \#, \lambda)$

- E is the set of event occurrences
- $\leq$ is the causality relation (a partial order)
- $\#$ is a binary conflict relation
  - Irreflexive, symmetric
- Conflict is inherited via causality
  - $e\#f$ and $f \leq f'$ implies $e\#f'$
- $\lambda : E \rightarrow \Sigma$ labels each event occurrence with an action
- Two events are concurrent if they are not related by $\leq$ or $\#$ — $e \; co \; f$

Trace event structures

Let (Σ, I) be a trace alphabet

$ES = (E, \leq, \#, \lambda)$ is a trace event structure if

- $e \#_\mu f \Rightarrow \lambda(e) \neq \lambda(f)$
  - Determinacy!
- If $e \lessdot f$ or $e \#_\mu f$, $(\lambda(e), \lambda(f)) \notin I$
- If $(\lambda(e), \lambda(f)) \notin I$ then $e \leq f$ or $f \leq e$ or $e \# f$.

Trace event structures

Let (Σ, I) be a trace alphabet

$ES = (E, \leq, \#, \lambda)$ is a trace event structure if

- $e \#_\mu f \Rightarrow \lambda(e) \neq \lambda(f)$
  - Determinacy!
- If $e \lessdot f$ or $e \#_\mu f$, $(\lambda(e), \lambda(f)) \notin I$
- If $(\lambda(e), \lambda(f)) \notin I$ then $e \leq f$ or $f \leq e$ or $e \# f$.

Fact

Any event structure constructed from the traces of a deterministic concurrent system is a trace event structure

Event structures as relational structures

Instead of temporal logics, consider

- First-Order Logic (FOL)
- (Variations of) Monadic Second Order logics (MSO)

FOL and MSO are logics over relational structures — a set with a collection of relations defined over the set

Labelled event structures give rise naturally to relational structures

- $ES = (E, \leq, \#, \lambda)$ labelled by $\Sigma = \{a_1, a_2, \ldots, a_n\}$
- Corresponding relational structure is $(E, \leq, \#, \ell_{a_1}, \ell_{a_2}, \ldots, \ell_{a_n})$
  - Each ℓ_{a_i} is a unary predicate such that $\ell_{a_i}(e)$ is true iff $\lambda(e) = a_i$

FOL and MSO

Relational structure $(E, \leq, \#, \ell_{a_1}, \ell_{a_2}, \dots, \ell_{a_n})$

- $\{x, y, \dots\}$: variables representing individual events
- $\{X, Y, \dots\}$: variables representing sets of events

FOL

$$x = y \mid x \leq y \mid x \# y \mid \ell_a(x) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\varphi(x)$$

MSOL

$$x = y \mid x \leq y \mid x \# y \mid \ell_a(x) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\varphi(x) \mid \exists X.\varphi(x)$$

The model-checking problem

- We are given a regular trace language L
  - Set of traces whose linearizations is a regular language
- From the prime traces, those with a single maximal event, we can extract an event structure ES_L
- Given a formula φ in FOL/MSO, does $ES_L \models \varphi$?

MSO over trace event structures is undecidable

[Walukiewicz]

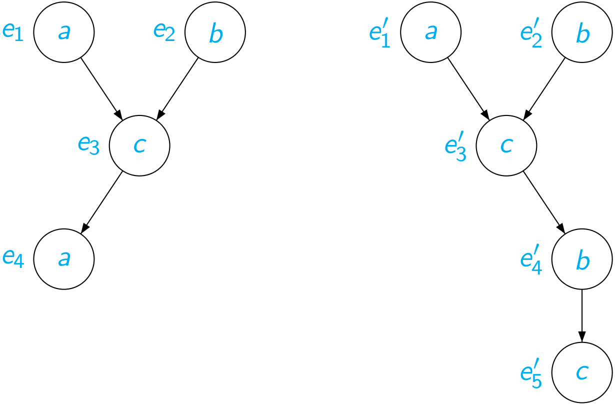
- Alphabet $\{a, b, c\}$ with $I = \{(a, b), (b, a)\}$
- Consider trace language generated by words of the form $a^* b^* c)$
- Each prime trace/event $[a^j b^k c]$ encodes a grid point (j, k)
- Set variables describe an assignment of colours to these events
- MSO can describe that this colouring/tiling of the grid is valid
- To get around this, restrict MSO to Monadic Trace Logic (MTL)
  - Quantify over conflict-free subsets of E

FOL over trace event structures is decidable

- Let $\varphi(x_1, x_2, \ldots, x_k)$ be an FOL formula
- φ defines a k-ary relation over events
  $R_\varphi = \{(e_1, e_2, \ldots, e_k) \mid ES \models \varphi(e_1, e_2, \ldots, e_k)\}$
- Recall that each event is actually a prime trace, so R_φ is a relation over traces in L
- Combine each tuple $(t_1, t_2, \ldots, t_k) \in R_\varphi$ into a single braided trace (over a new alphabet)
- Model-checking R_φ is equivalent to checking that the set of braided traces corresponding to R_φ is non-empty
- For each formula φ, the braided traces corresponding to R_φ form a regular trace language

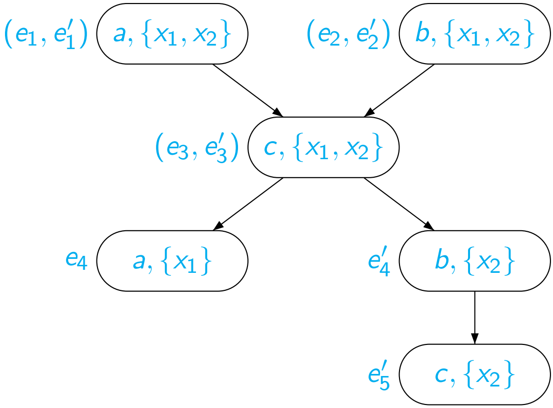
Braiding traces

Overlap traces as far as possible, recording for each overlapped event, which components participate in that event



Braiding traces

Overlap traces as far as possible, recording for each overlapped event, which components participate in that event



Braiding traces . . .

- Braided traces over new alphabet Σ_B with symbols (a, Y) where
  - $a \in \Sigma$ is a letter from the original alphabet
  - $Y \subseteq \{x_1, x_2, \ldots, x_k\}$
- $((a, X), (b, Y)) \in I_B$ if $(a, b) \in I$ or $X \cap Y = \emptyset$

Braiding traces . . .

- Braided traces over new alphabet Σ_B with symbols (a, Y) where
  - $a \in \Sigma$ is a letter from the original alphabet
  - $Y \subseteq \{x_1, x_2, \ldots, x_k\}$
- $((a, X), (b, Y)) \in I_B$ if $(a, b) \in I$ or $X \cap Y = \emptyset$

Observation

- If $(a, X) \leq (b, Y)$ in a braided trace, then $Y \subseteq X$
  - The second component monotonically decreases along each chain of dependent letters
- This property can be checked by a finite-state automaton

Braiding traces . . .

Theorem

For each FOL formula $\varphi(x_1, x_2, \ldots, x_k)$, the corresponding braided trace language is regular

Proof

By induction on the structure of φ

Braiding traces . . .

φ is $x = y$

Braiding traces . . .

φ is $x = y$

- $(t1, t2) \in R_\varphi$ iff $t_1 = t_2$
- Braided trace is isomorphic to t_1 (and t_2)
- Each action is labelled $\{x_1, x_2\}$

Braiding traces ...

φ is $x = y$

- $(t1, t2) \in R_\varphi$ iff $t_1 = t_2$
- Braided trace is isomorphic to t_1 (and t_2)
- Each action is labelled $\{x_1, x_2\}$
- Check that projection onto Σ is a prime trace in L
  - Note: If L is a regular trace language, the prime traces of L also form a regular trace language
- Check that second component of each label is $\{x_1, x_2\}$

Braiding traces . . .

φ is $x \leq y$

Braiding traces . . .

φ is $x \leq y$

- $(t1, t2) \in R_\varphi$ iff t_2 extends t_1
- Braided trace is isomorphic to t_2
- Each action is labelled $\{x_1, x_2\}$ or $\{x_2\}$

Braiding traces . . .

φ is $x \leq y$

- $(t1, t2) \in R_\varphi$ iff t_2 extends t_1
- Braided trace is isomorphic to t_2
- Each action is labelled $\{x_1, x_2\}$ or $\{x_2\}$
- Check that projection onto Σ is a prime trace in L
- Check that second component of each label is $\{x_1, x_2\}$ or $\{x_2\}$
- Check that second component decreases monotonically along each chain of dependent letters

Braiding traces . . .

φ is $x\#y$

Braiding traces . . .

φ is $x\#y$

- $(t1, t2) \in R_\varphi$ iff t_1 and t_2 diverge
- At least one action each labelled only $\{x_1\}$ and $\{x_2\}$
- Braided trace restricted to
  - actions labelled $\{x_1, x_2\}$ or $\{x_1\}$ is isomorphic to t_1
  - actions labelled $\{x_1, x_2\}$ or $\{x_2\}$ is isomorphic to t_2

Braiding traces ...

φ is $x\#y$

- $(t1, t2) \in R_\varphi$ iff t_1 and t_2 diverge
- At least one action each labelled only $\{x_1\}$ and $\{x_2\}$
- Braided trace restricted to
  - actions labelled $\{x_1, x_2\}$ or $\{x_1\}$ is isomorphic to t_1
  - actions labelled $\{x_1, x_2\}$ or $\{x_2\}$ is isomorphic to t_2
- Check that projections $\{x_1, \ldots\}$ and $\{x_2, \ldots\}$ are both prime traces in L
- Check that there is at least one event each with second component of label $\{x_1\}$ and $\{x_2\}$
- Check that second component decreases monotonically along each chain of dependent letters

Braiding traces . . .

φ is $\exists y.\psi(y, x_1, \ldots, x_k)$

- By induction hypothesis, braided trace language for R_ψ is regular
- Define a natural projection operator to eliminate y from a set of braided traces
  - Project onto $(x_1, \ldots, x_k) \Rightarrow$ drop y from each event's label
  - Erase any event whose initial label was $\{y\}$ (and hence now has an empty label)
- If B is a regular language of braided traces over variables $\bar{x}$, its projection onto any subset of $\bar{x}$ is also regular
- Braided trace language for φ is obtained by projecting the language for ψ onto $(x_1, x_2, \ldots, x_k)$

Braiding traces ...

φ is $\neg\psi$: Easy

φ is $\psi_1 \wedge \psi_2$

- $\psi_1(x_1, \ldots, x_k)$ and $\psi_2(y_1, \ldots, y_m)$ so braided traces for φ are over $(x_1, \ldots, x_k, y_1, \ldots, y_m)$
- In general, some variables overlap between ψ_1, ψ_2
$$\varphi(\bar{x}, \bar{y}, \bar{z}) = \psi_1(\bar{x}, \bar{z}) \wedge \psi_2(\bar{y}, \bar{z})$$
- Define an "expansion" operator:
  - B, a set of braided traces over $\bar{u} = (u_1, u_2, \ldots, u_k)$
  - $\bar{v} = (v_1, v_2, \ldots, v_m)$, a new set of variables
  - $B \uparrow \bar{v}$: all braided traces over $(\bar{u}, \bar{v}) = (u_1, \ldots, u_k, v_1, \ldots, v_k)$ whose projection onto $\bar{u}$ lies in B.
- Then, the language for φ is $(B_{\psi_1} \uparrow \bar{y}) \cap (B_{\psi_2} \uparrow \bar{x})$

MTL

- MTL is MSO with set quantifiers restricted to conflict-free subsets of E
- In FOL proof, each individual variable x is assigned an event e, which can be regarded as a prime trace
- Can we represent conflict-free subsets of E as traces?

MTL

- MTL is MSO with set quantifiers restricted to conflict-free subsets of E
- In FOL proof, each individual variable x is assigned an event e, which can be regarded as a prime trace
- Can we represent conflict-free subsets of E as traces?

- If $X \subset E$ is conflict-free, so is $\downarrow X$
- Thus, $\downarrow X$ is a trace (not necessarily prime)
- Not all events in $\downarrow X$ are part of the subset
  - Add a tag from $\{\perp, \top\}$ to indicate which events in $\downarrow X$ belong to X and which do not

MTL

- MTL is MSO with set quantifiers restricted to conflict-free subsets of E
- In FOL proof, each individual variable x is assigned an event e, which can be regarded as a prime trace
- Can we represent conflict-free subsets of E as traces?

- If $X \subset E$ is conflict-free, so is $\downarrow X$
- Thus, $\downarrow X$ is a trace (not necessarily prime)
- Not all events in $\downarrow X$ are part of the subset
  - Add a tag from $\{\bot, \top\}$ to indicate which events in $\downarrow X$ belong to X and which do not

- Can again assign a set of braided traces with each formula φ
- Show by induction on φ that this set is regular

In perspective

FOL over traces can express all natural temporal modalities

- $ES, e \models A_{\leq}\varphi$ if at every f such that $e \leq f$, $ES, f \models \varphi$
- $ES, e \models E_{\#}\varphi$ if there exists f such that $e \# f$ and $ES, f \models \varphi$
- $ES, e \models A_{co}\varphi$ if at every f such that $e \; co \; f$, $ES, f \models \varphi$
- ...

In perspective

FOL over traces can express all natural temporal modalities

- $ES, e \models A_{\leq}\varphi$ if at every f such that $e \leq f$, $ES, f \models \varphi$
- $ES, e \models E_{\#}\varphi$ if there exists f such that $e\#f$ and $ES, f \models \varphi$
- $ES, e \models A_{co}\varphi$ if at every f such that $e\ co\ f$, $ES, f \models \varphi$
- ...

In one shot, decidability of FOL over trace event structures shows that all (reasonable) temporal logics are decidable!

In perspective

FOL over traces can express all natural temporal modalities

- $ES, e \models A_{\leq}\varphi$ if at every f such that $e \leq f$, $ES, f \models \varphi$
- $ES, e \models E_{\#}\varphi$ if there exists f such that $e \# f$ and $ES, f \models \varphi$
- $ES, e \models A_{co}\varphi$ if at every f such that $e\ co\ f$, $ES, f \models \varphi$
- ...

In one shot, decidability of FOL over trace event structures shows that all (reasonable) temporal logics are decidable!

What more remains to be done?

In perspective . . .

- System is presented as a regular trace language
- Implicitly, we assume a deterministic machine recognizing the language
- Model-checking is typically applied to a given system model
  - May be nondeterministic
  - Distinction between labelled and unlabelled systems in models like Petri nets
- What is the status of branching-time model-checking for labelled concurrent systems?

In perspective . . .

- In sequential systems, model-checking is intimately connected to automata theory
  - Tree automata
  - Alternating automata (on strings and trees)
- In concurrent systems, the theory of “string” automata is reasonably well-understood
  - Asynchronous automata, Zielonka’s theorem
- How do we define alternating automata on traces?