Model transformations and Properties/Equivalence reductions (Ch. 4-5)

Vesa Luukkala

29. October 2001

Outline

- Introduction
- Model transformations and properties
  - linear time world
  - safety properties
  - simulation and the properties it preserves
- Equivalence reductions
  - bisimulation and the properties it preserves
  - distinguishing power and expressiveness
  - Ehrenfeucht-Fraïsse games
  - autobisimulations and bisimulation minimization

Introduction

- problem is to verify properties $M \models \varphi$
- to ease the problem reduce the size of the model in a methodical way
- reductions that preserve certain classes of properties are identified
- in linear time (LTL) case it is possible to transfer logic formulas to automatons and vice versa
- in both linear and branching time cases it is possible to choose minimization relations so that expressable properties are preserved

Models, automata and transition systems (1/3)

- a structural level connection between ω-automata and LTL formulas - construct an automata from a Kripke model
- Kripke model $\mathcal{M} = (U, \mathcal{I}, w_0)$ with predicates from $\mathcal{P}$ and accessibility relations from $\mathcal{R}$
- alphabet $\Sigma = 2^{\mathcal{P}} \times \mathcal{R}$, an ω-word $\sigma = \sigma_0\sigma_1\ldots$, where $\sigma_i = (a_i, R_i)$
- σ is generated from $\mathcal{M}$ if there is a mapping ρ from indices of σ to points of U
  - $\rho(0) = w_0$, (initial states match)
  - if $\rho(i) = w$, then $a_i = \mathcal{L}(w)$, (predicates match)
  - if $\rho(i) = w$ and $\rho(i+1) = w'$ then $(w, w') \in \mathcal{I}(R_i)$, (transition relations match)
  - if σ is finite with last letter σ_n and $\rho(n) = w$, then w is terminal (generated words represent maximal paths in the model)
- the *language generated by* $\mathcal{M}$ is the set of ω-words generated by $\mathcal{M}$
- Kripke-models can be expressed as weakly fair (all states recurring, terminals accepting) transition systems with alphabet Σ

Models, automata and transition relations (2/3)

- models can be seen as automata (by lemma 4.1), also for every **LTL** formula there exists a Büchi-automaton
- φ is an **LTL** formula and $\mathcal{M}$ (with single accessibility relation)
- transform $\mathcal{M}$ to weakly fair transition system $\mathcal{M}_{\mathcal{A}}$ and φ to Büchi-automaton $\mathcal{M}_{\varphi}$
- φ is *sequence valid* in $\mathcal{M}$ iff the language generated by $\mathcal{M}_{\mathcal{A}}$ is subset of the language accepted by $\mathcal{M}_{\varphi}$:

$$\mathcal{M} \models \varphi \text{ iff } L(\mathcal{M}_{\mathcal{A}}) \subseteq L(\mathcal{M}_{\varphi})$$

- or $L(\mathcal{M}_{\mathcal{A}}) \cap \overline{L(\mathcal{M}_{\varphi})} = \{\}$ or $L(\mathcal{M}_{\mathcal{A}}) \times L(\mathcal{M}_{\varphi}) = \{\}$
- model checking problem is turned to nonemptiness check of the Büchi-automaton

Models, automata and transition relations (3/3)

- the product automaton $\mathcal{M}_{\mathcal{A}} \times \mathcal{M}_{\varphi}$ must accept an infinite word σ iff both component automatons do - the inifinite run must visit the recurring states of both components infinite often
- usually φ is transformed to $\mathcal{M}_{\neg\varphi}$ and model checking consists of checking that $L(\mathcal{M}_{\mathcal{A}} \times \mathcal{M}_{\neg\varphi})$ is empty
- since both $\mathcal{M}$ and φ can be represented as an automaton, φ can be regarded as an abstract version of the "implementation" $\mathcal{M}$, thus $\mathcal{M}_I \models \mathcal{M}_S$ if $L(\mathcal{M}_I) \subseteq L(\mathcal{M}_S)$

  Theorem 4.2
  $\mathcal{M}_1, \mathcal{M}_2$ are Büchi-automatons:
  - $\mathcal{M}_1 \models \mathcal{M}_2$ iff for all properties φ, if $\mathcal{M}_2 \models \varphi$ then $\mathcal{M}_1 \models \varphi$
  - $\mathcal{M}_1 \models \mathcal{M}_2$ iff for all ω-regular φ, if $\mathcal{M}_2 \models \varphi$ then $\mathcal{M}_1 \models \varphi$
- to prove $\mathcal{M}_1 \models \varphi$, create a smaller $\mathcal{M}_2$: $\mathcal{M}_1 \models \mathcal{M}_2$ and check $\mathcal{M}_2 \models \varphi$

Safety and Liveness Properties (1/4)

- for natural models $\mathcal{M}^{[..i]}$ is the model consisting of first i points of $\mathcal{M}$, $\mathcal{M} \circ \mathcal{M}'$ is the concatenation of both models ($\mathcal{M}$ if $\mathcal{M}$ is infinite)
- φ is a *safety property*, iff for all natural models $\mathcal{M}$,

$$\mathcal{M} \models \varphi \text{ if } \forall i \exists \mathcal{M}' : \mathcal{M}^{[..i]} \circ \mathcal{M}' \models \varphi$$

  if the safety property is broken, there must be a finite prefix that can not be completed to an accepting computation

- φ is a *liveness property*, iff for all natural models $\mathcal{M}$,

$$\forall i \exists \mathcal{M}' : \mathcal{M}^{[..i]} \circ \mathcal{M}' \models \varphi$$

Theorem 4.3 (Properties of safety and liveness properties)

- safety props are closed under finite unions and arbitrary intersections
- liveness props are closed under finite unions but not under intersections
- $\top$ is the only prop that is both safety and liveness
- for any property φ thre exists a safety property φ_S and a liveness property φ_L s.t. $\varphi = (\varphi_S \cap \varphi_L)$

Safety Properties (2/4)

- a syntactical definition of an **LTL** safety property:

  Theorem 4.4
  Every temporal formula built from literals with $\bot, \top, \wedge, \vee, \mathbf{W}^+$ defines a safety property.

  an alternative characterization would be via past temporal formulas: $\mathbf{G}^*\psi$

Safety Properties; characterization by automatons (3/4)

- a binary relation $\Delta \subseteq U \times U$ is *image finite* if for any $x \in U$ the set $\{y \in U \bar{(} x, y) \in \Delta\}$ is finite – "every state has finite number of successors"
- transition system S, Δ, S_0 is *finitary* if S_0 is finite and Δ is image finite – "only finite nondeterminism allowed"

  Theorem 4.5
  Any finitary transition system defines a safety property.

- the finitary requirement prevents the following example that defines $(\mathbf{F}^*\mathbf{X}\bot)$ (all finite strings)

  Theorem 4.6
  For every ω-regular safety property there is a finite transition system defining this property.

- there exists a tableau procedure (section 7) for obtaining a daterministic transition system for **LTL** safety properties

Safety Properties (in practice) (4/4)

- to check that a model sequence-validates an ω-regular safety property can be checked by the language containment problem $\mathcal{M} \models \varphi$ iff $L(\mathcal{M}_{\mathcal{A}}) \subseteq L(\mathcal{M}_{\varphi})$
- this can be checked by executing $\mathcal{M}_{\mathcal{A}}$) and $\mathcal{M}_{\varphi}$ concurrently in lock-step (can be used in specification-based testing)
- for finitary transition systems it is sufficient to check whether $\mathcal{M}_2 \models \varphi$ implies $\mathcal{M}_1 \models \varphi$ for all safety properties φ to establish $\mathcal{M}_1 \models \mathcal{M}_2$:

Theorem 4.7
$\mathcal{M}_1, \mathcal{M}_2$ are finitary transition systems. $\mathcal{M}_1 \models \mathcal{M}_2$ iff for all safety properties φ, if $\mathcal{M}_2 \models \varphi$ then $\mathcal{M}_1 \models \varphi$.

(variation of 4.2 that was for all ω-regular properties)

Simulation relations (1/6)

- weaker preorders than the language inclusion are useful
  - language containment for large nondeterministic systems is hard
  - it may be useful to formulate properties regarding the structure of the system
- $\mathcal{M}_1$ is a *submodel* of a model $\mathcal{M}_2$ ($\mathcal{M}_1 \sqsubseteq \mathcal{M}_2$) if
  - $U_1 \subseteq U_2$
  - $\mathcal{I}_1 = \mathcal{I}_2 \downarrow U_1$
  - $w_1 = w_2$

  "part of a bigger graph"
- *generated submodel* is the model consisting of all reachable states; preserves all temporal properties
- in general it is usually better idea to combine states rather than delete them

Simulation relations (2/6)

- for models $\mathcal{M}_1$ and $\mathcal{M}_2$, a relation $H \subseteq U_1 \times U_2$ is a *simulation* ($\mathcal{M}_1 \rightrightarrows \mathcal{M}_2$) if
  - $(w_1, w_2) \in H$
  - $\forall p \in \mathcal{P}, u \in U_1, v \in U_2$ if $(u, v) \in H$ then $u \in \mathcal{I}_1(p)$ iff $v \in \mathcal{I}_2(p)$
  - $\forall u, v : (u, v) \in H$ and for all R, u' s.t. $(u, u') \in \mathcal{I}_1(R)$ there is a v' s.t. $(v, v') \in \mathcal{I}_2(R)$ and $(u', v') \in H$

  To have a simulation $\mathcal{M}_1 \rightrightarrows \mathcal{M}_2$ ($\mathcal{M}_2$ simulates $\mathcal{M}_1$), $\mathcal{M}_2$ must be able to do the same transition as $\mathcal{M}_2$ – one state of $\mathcal{M}_2$ can simulate several $\mathcal{M}_1$ states
- $\mathcal{M}_2$ is an abstraction of $\mathcal{M}_1$, less states but more behaviours
- simulation is a preorder on class of all models (4.8)
- other properties:
  - if $\mathcal{M}_1 \sqsubseteq \mathcal{M}_2$ then $\mathcal{M}_1 \rightrightarrows \mathcal{M}_2$
  - if $\mathcal{M}_1 \rightrightarrows \mathcal{M}_2$ then $\mathcal{M}_1 \models \mathcal{M}_2$
  - for deterministic models $\mathcal{M}_1 \models \mathcal{M}_2$ iff $\mathcal{M}_1 \rightrightarrows \mathcal{M}_2$ (a model is *deterministic* if for every $w \in U$ and $R \in \mathcal{R}$ there exists at most one $w' \in U$ s.t. $(w, w') \in \mathcal{I}(R)$)

Simulation relations: preserved properties (3/6)

- if there is a simulation relation between the models, then the models have a simulation relation (4.9)
- a *modal box formula* is a formula without the diamond operator ("eventually")
  - literals and $\top, \bot$
  - if φ, ψ are modal box formulas, then $(\varphi \wedge \psi), (\varphi \vee \psi), [R]\varphi$ are modal box formulas

Theorem 4.8
Let $\mathcal{M}_1, \mathcal{M}_2$ be Kripke-models. $\mathcal{M}_1 \rightrightarrows \mathcal{M}_2$ implies that for all modal box formulas φ, if $\mathcal{M}_2 \models \varphi$ then $\mathcal{M}_1 \models \varphi$.

like 4.2 and 4.6, this allows checking the modal box formula of a smaller model

Simulation relations: preserved properties (4/6)

- simulation can maintain more expressive logics
- a **ACTL** formula is a **CTL** formula without the **E** quantifier
  - literals and $\top, \bot$
  - if φ, ψ are **ACTL** formulas, then $(\varphi \wedge \psi), (\varphi \vee \psi), \mathbf{A}(\varphi \mathbf{U}^{+} \psi), \mathbf{A}(\varphi \mathbf{W}^{+} \psi)$ are **ACTL** formulas
- **ACTL** formulas describe properties that are valid in all paths of the model, singling out properties of one path is not possible

  Theorem 4.9
  Let $\mathcal{M}_1, \mathcal{M}_2$ be Kripke-models. $\mathcal{M}_1 \rightrightarrows \mathcal{M}_2$ implies that for all **ACTL** formulas φ, if $\mathcal{M}_2 \models \varphi$ then $\mathcal{M}_1 \models \varphi$.

- converse does not hold for non-finitary system (see counterexample for modally indistinguishable models)

Simulation relations: preserved properties (5/6)

- the example can not be distinguished by any modal formula:

  Theorem 4.10
  For any $\varphi \in \mathbf{ML}$ it holds that $\mathcal{M}_1 \models \varphi$ iff $\mathcal{M}_2 \models \varphi$.

- although by above for all modal box formulas if $\mathcal{M}_2 \models \varphi$ then $\mathcal{M}_1 \models \varphi$, there is no simulation relation between $\mathcal{M}_1, \mathcal{M}_2$ (as opposed to 4.8)

- image finite cases the converse of 4.8 holds

  Theorem 4.11
  Let $\mathcal{M}_1, \mathcal{M}_2$ be image finite Kripke-models. $\mathcal{M}_1 \rightrightarrows \mathcal{M}_2$ iff for all modal box formulas φ, if $\mathcal{M}_2 \models \varphi$ then $\mathcal{M}_1 \models \varphi$.

Simulation relations: algorithm (6/6)

- for deterministic finite automata there are efficient algorithms for language inclusion
- an algorithm for creating a simulation relation $H = U_1 \times U_2$ between two nondeterministic models:
  1. place all pairs of states with matching properties into the first iteration of the relation H^0
  2. for the next iteration, place a pair of H^n to H^{n+1} if the model to be simulated has a transition that the simulating model can match – this simulated transition should end to some other pair in H^n
- since this is a finite model, eventually $H^n = H^{n+1}$
- intersection of all H^n is the largest simulation relation

Bisimulations (1/14)

- equivalence is a symmetric preorder
- a preorder $\preceq$ can induce an equivalence $\simeq$: $\mathcal{M}_1 \simeq \mathcal{M}_2$ iff $\mathcal{M}_1 \preceq \mathcal{M}_2$ and $\mathcal{M}_2 \preceq \mathcal{M}_1$
- submodel ordering $\sqsubseteq$ induces isomorphism, sequence validness $\models$ induces equivalence of the generated languages
- bisimulation $\rightleftarrows$ is an equivalence relation between universes of two Kripke-models $\mathcal{M}_1, \mathcal{M}_2$:
  - $w_1 \rightleftarrows w_2$
  - if $u \rightleftarrows v$ then $u \in \mathcal{I}_1(p)$ iff $v \in \mathcal{I}_2(p)$
  - if $u \rightleftarrows v$ and $(u, u') \in \mathcal{I}_1(R)$ then there exists v' s.t. $(v, v') \in \mathcal{I}_2(R)$ and $u' \rightleftarrows v'$
  - if $u \rightleftarrows v$ and $(v, v') \in \mathcal{I}_2(R)$ then there exists u' s.t. $(u, u') \in \mathcal{I}_1(R)$ and $u' \rightleftarrows v'$

Bisimulations (2/14)

- Some properties of bisimilar models
  - each model is bisimilar to one where duplicate states with same inputs and outputs are removed
  - a model is bisimilar to its reachable part
  - a model is bisimilar to its unfolding
- if $\mathcal{M}_1 \rightleftarrows \mathcal{M}_2$ then $\mathcal{M}_1 \rightrightarrows \mathcal{M}_2$ and $\mathcal{M}_2 \rightrightarrows \mathcal{M}_1$, but not necessarily the other way around
- models $\mathcal{M}_1, \mathcal{M}_2$ are *equivalent w.r.t. to logic* $\mathbf{L}$ ($\mathcal{M}_1 \equiv_{\mathbf{L}} \mathcal{M}_2$) if for all well formed formulas of $\mathbf{L}$ it holds that $\mathcal{M}_1 \models \varphi$ iff $\mathcal{M}_2 \models \varphi$
- the relation $\equiv_{\mathbf{FOL}}$ is the elementary equivalence

Bisimulations: preserved properties (3/14)

- Bisimulation relations are precisely those equivalences that preserve all modal formulas (compare to modal box preservation of simulations (4.10)) – modal formulas are bisimulation invariant:

  Theorem 5.2
  Bisimilar models are modally equivalent: if $\mathcal{M}_1 \rightleftarrows \mathcal{M}_2$ then $\mathcal{M}_1 \equiv_{\mathbf{ML}} \mathcal{M}_2$.

- converse requires image finiteness:

  Theorem 5.3
  Image finite models are modally equivalent iff they are bisimilar: if $\mathcal{M}_1, \mathcal{M}_2$ are image finite, then $\mathcal{M}_1 \rightleftarrows \mathcal{M}_2$ iff $\mathcal{M}_1 \equiv_{\mathbf{ML}} \mathcal{M}_2$.

Bisimulations: preserved properties (4/14)

- by restricting the model to finite Kripke-models, it is possible have similar resuls for more expressive logics:

  Theorem 5.4
  Let $\mathcal{M} \triangleq (U, \mathcal{I}, w)$ be a finite model ($|U| = n$), and let φ be a monotonic μ**TL**-formula. Then $\mathcal{M} \models \nu q\, \varphi$ iff $\mathcal{M} \models \nu^n q\, \varphi$.

- since modal logic is a sublanguage of μ**TL**:

  Theorem 5.5
  Finite models are monotonic μ**TL**-equivalent iff they are bisimilar: if $\mathcal{M}_1, \mathcal{M}_2$ are finite, then $\mathcal{M}_1 \rightleftarrows \mathcal{M}_2$ iff $\mathcal{M}_1 \equiv_{\mu\mathbf{TL}} \mathcal{M}_2$.

Bisimulations: distinguishing power (5/14)

- it is possible to user weaker logics to distinguish between models:

  Theorem 5.6
  Finite models are monotonic μ**TL**-equivalent iff they are bisimilar: if $\mathcal{M}_1, \mathcal{M}_2$ are finite, then $\mathcal{M}_1 \rightleftarrows \mathcal{M}_2$ iff $\mathcal{M}_1 \equiv_{\mathbf{ML}} \mathcal{M}_2$.

- if two finite models can be distinguished by a formula of logic **CTL*** then they can be distinguished by a **CTL** formula as well – **CTL*** can be transferred to **MSOL** and thus μ**TL**
- logics with different expressiveness can have the same distinguishing capabilitites

Bisimulations: expressiveness and distinguishing power (6/14)

- logic **L2** is *at least as expressive as* **L1** iff for any formula $\varphi_1 \in \mathbf{L1}$ there exists a formula $\varphi_2 \in \mathbf{L2}$ s.t. for all models $\mathcal{M}$: $\mathcal{M} \models \varphi_1$ iff $\mathcal{M} \models \varphi_2$
- **L1**, **L2** *have the same expressive power* if **L1** is at least as expressive as **L2** and vice versa – for each formula in one logic there is an equivalent one in the second
- logic **L2** is *at least as distinguishing as* **L1** if any two models that are inequivalent w.r.t. **L1** are inequivalent w.r.t. **L2** – or iff $\mathcal{M}_1 \equiv_{\mathbf{L2}} \mathcal{M}_2$ implies $\mathcal{M}_1 \equiv_{\mathbf{L1}} \mathcal{M}_2$
- **L1**, **L2** *have the same distinguishing power* if **L1** is at least as distinguishing as **L2** and vice versa – or iff for all models it holds that $\mathcal{M}_1 \equiv_{\mathbf{L2}} \mathcal{M}_2$ iff $\mathcal{M}_1 \equiv_{\mathbf{L1}} \mathcal{M}_2$

Bisimulations: expressiveness and distinguishing power (7/14)

- expressiveness is a finer equivalence relation than distinguishability

  Theorem 5.7
  If **L1** is at most as expressive as **L2**, then it is at most as distinguishing. If **L1** and **L2** have the same expressive power, then they have the same distinguishing power but not vice versa.

Bisimulations: yardstick for expressiveness (8/14)

- any formula φ is *preserved under bisimulations* if for all models $\mathcal{M}_1 \rightleftarrows \mathcal{M}_2$ it holds that $\mathcal{M}_1 \models \varphi$ iff $\mathcal{M}_2 \models \varphi$
- a logic **L** is *bisimulation invariant* if all well formed formulas of **L** are preserved under bisimulations
- multimodal logics are bisimulation invariant (5.2), but this holds for more expressive logics like μ**TL**:

  Theorem 5.8
  If $\mathcal{M}_1 \rightleftarrows \mathcal{M}_2$ then for any positive μ**TL** formula φ it holds that $\mathcal{M} \models \varphi_1$ iff $\mathcal{M} \models \varphi_2$.

- the reverse direction provides a connection between bisimulations, first order and model expressiveness - specifies which 1st order formulas can be transferred to **ML**

  Theorem 5.9 (Expressive completeness of ML)
  For any 1st order formula φ (with 1 free variable) which is preserved under bisimulations there exists an equivalent multimodal formula.

Bisimulations: yardstick for expressiveness (9/14)

- the same result can be extended to 2nd order formulas and μ**TL**

  Theorem 5.10 (Expressive completeness of μTL)
  Let φ be any **MSOL** property. Then φ is preserved under bisimulations iff φ is definable by positive μ**TL** formula.

- every logic which is bisimulation invariant and has a semantical translation to **MSOL** can be translated to μ**TL**

Bisimulations: Ehrenfeucht-Fraïsse games (10/14)

- a convinient way of imagining bisimulations (and equivalences w.r.t. other logics)
- two players: Ann and Bob, each having an unlimited number of identified pieces: $a_0, a_1, \ldots$ and $b_0, b_1, \ldots$
- game is played on two Kripke-structures, both place their first pieces on different models – labels must match or Bob loses
- Ann places her $(i+1)$th piece on either of the boards honouring the transition relation w.r.t placed pieces
- Bob has to match Anns move on the other board by locating the ith piece on that board and placing the piece honouring the transition relation
- if Bob can not match Anns move he loses, if he can play the game forever he wins

Bisimulations: Ehrenfeucht-Fraïsse games ($10\frac{1}{2}/14$)

- Ann *can force a win within n rounds* if she can place her piece s.t. Bob loses immediately or after $n - 1$ rounds
- Ann has a *winning strategy* if there is n s.t. she can force a win – Bob has a winning strategy if Ann does not

  Theorem 5.11
  Ann has a winning strategy iff the two models are not bisimilar; Bob has a winning startegy iff they are bisimilar.

- allowing sets of pieces Ann has a winnign strategy iff the boards can be distinguished by **MSOL** formula

Bisimulations: auto-bisimulations (11/14)

- to minimize a Kripke model w.r.t. bisimulations
- note that all definitions have not forbidden bisimulations to points in the same model: *auto-bisimulations*

  Theorem 5.12
  The union of any number of auto-bisimulations on a model is again an auto-bisimulations.

  Thus the greatest auto-bisimulation is the union of all auto-bisimulations in the model.
- for each auto-bisimulation there exists greatest equivalence relation $\equiv$ that includes the auto-bisimulation ($\rightleftarrows \subseteq \equiv$) and is also an auto-bisimulation

Bisimulations: auto-bisimulations (12/14)

- for any model $\mathcal{M} \triangleq (U, \mathcal{I}, w_0)$ and equivalence relation $\equiv$ on U *quotient of* $\mathcal{M}$ *w.r.t* $\equiv$ is the model $\mathcal{M}^{\equiv} \triangleq (U^{\equiv}, \mathcal{I}^{\equiv}, w_0^{\equiv})$ s.t.
  - $U^{\equiv}$ is the set of equivalence classes of U w.r.t. $\equiv$
  - $w_0^{\equiv}$ is the equivalence class of w_0
  - $\mathcal{I}^{\equiv}$:
    * $w^{\equiv} \in \mathcal{I}^{\equiv}(p)$ if there is $w \in w^{\equiv}$ s.t. $w \in \mathcal{I}(p)$
    * $(w_1^{\equiv}, w_2^{\equiv}) \in \mathcal{I}^{\equiv}(R)$ if there are $w_1 \in w_1^{\equiv}$ and $w_2 \in w_2^{\equiv}$ s.t. $(w_1, w_2) \in \mathcal{I}(R)$

  Theorem 5.13
  If the equivalence relation $\equiv$ is an auto-bisimulation, then $\mathcal{M} \rightleftarrows \mathcal{M}^{\equiv}$.

- the quotient of the model w.r.t. its largest autobisimulation is the minimal representation of the model

Bisimulations: partitions (13/14)

- for any set of points $P \subseteq U$

$$< R > P = \{w | \exists w' \in P, (w, w') \in \mathcal{I}(R)\}$$

  those nodes that have a transition to P.

- given a partition U to equivalence classes, a component $w^{\equiv}$ is *uniform* if

$$\forall p \in P : w^{\equiv} \subseteq \mathcal{I}(p) \quad \vee \quad w^{\equiv} \cap \mathcal{I}(p) = \{\}$$

  the nodes in partition have the same labeling (propositions).

- a component $w^{\equiv}$ is *stable* w.r.t set P if

$$\forall R : w^{\equiv} \subseteq < R > P \quad \vee \quad w^{\equiv} \cap < R > P = \{\}$$

  it is possible to access P from partition

- a partition is *stable* if all components are uniform and stable w.r.t other partitions

  Theorem 5.14
  The coarsest stable partition is the largest auto-bisimulation.

Bisimulations: algorithm (14/14)

- To construct the coarsest stable partition:
  - start with a trivial partition of one component
  - repeat until no new partitions are created and choose nondeterministically
    - 1. choose a component $w_0^{\equiv}$ and a proposition $p \in \mathcal{P}$
      2. split $w_0^{\equiv}$ to two uniform partitions in which the other partition has property p
    - 1. choose components $w_0^{\equiv}, w_1^{\equiv}$ and $R \in \mathcal{R}$
      2. split $w_0^{\equiv}$ to be stable w.r.t. $w_1^{\equiv}$ to those that have a transition to $w_1^{\equiv}$ and to those that have not
- Paige-Tarjan computes the same in $\mathcal{O}(m \cdot \log n)$, where n is number of points in model and m is the number of partitions in the result