29, October 2001

Vesa Luntakala

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

- Bisimulation and bisimulation minimization
- Ehrenfeucht-Fraïssé Games
- Distinctive power and expressiveness
- Bisimulation and the properties it preserves
- Equivalence reductions
- Simulation and the properties it preserves
- Safety properties
- Linear time world
- Model transformations and properties
- Introduction
relations so that expressible properties are preserved
in both linear and branching time cases, it is possible to choose mini-
and vice versa
mizations

in linear time (LTL) case, it is possible to transfer logical formulas to automata

reductions that preserve certain classes of properties are identical

to ease the problem, reduce the size of the model in a methodical way

\( \phi \models M \)

Introduction
accepting transition systems with alphabet X

Kripke models can be expressed as weakly fair (all states reachable, terminating, terminal)

The language generated by \( W \) is the set of m-words generated by W

words represent maximal paths in the model

If \( \omega \) is finite with last letter \( \alpha \) and \( m = (\alpha)\cdot \), then \( m \) is a terminal (generated)

(match)

\( (\alpha^i \cdot \nu) \in (\alpha^i \cdot m) \) then \( m = (1 + i)\cdot \alpha \) and \( m = (i)\cdot \alpha \) if \( \alpha \)

(match)

\( (\alpha^i) \alpha \in (\alpha^i) \alpha \) then \( \alpha = (1 + i)\cdot \alpha \) if \( \alpha \)

(initial states match)

\( \omega \) is generated from \( \omega \) if there is a mapping from indices of \( \omega \) to points of \( \Omega \)

\( (\alpha^i \cdot \nu) \in (\alpha^i \cdot m) \) then \( \nu = (1 + i)\cdot \nu \), where \( \nu \)

\( \alpha \) is a word in \( m \)-word alphabet of \( \alpha \) and \( m \)-word alphabet of \( \alpha \)

Y

From

Kripke model from a Kripke model construct an automata from a Kripke model

- a structural level connection between k-automata and LTL formulas

Models, automata and transition systems (1/3)
Büchi-automaton

model checking problem is turned to nonemptiness check of the

\[
\emptyset = (\phi \mathcal{W})^T \times (\forall \mathcal{W})^T \text{ or } \emptyset = (\phi \mathcal{W})^T \cup (\forall \mathcal{W})^T
\]

or •

\[(\phi \mathcal{W})^T \supset (\forall \mathcal{W})^T \text{ iff } \phi \models \mathcal{W}
\]

language accepted by \(\phi\) •

is sequence valid in \(W\) iff the language generated by \(\forall \mathcal{W}\) is subset of the

\(\phi\) •

transform to weakly fair transition system \(\forall \mathcal{W}\) and to Büchi-automaton

\(\phi\) •

is an \(\mathit{LTL}\) formula and \(\phi\) •

there exists a Büchi-automaton

Models, automata and transition relations (2/3)
\( \phi \models W \models \psi \) and check \( W \models I \) and check

- \( \phi \models W \models I \) if for all \( \phi \)-regular \( \phi \models W \models I \) if for all \( \phi \)-regular
- \( \phi \models W \models I \) if for all \( \phi \)-regular

\textbf{Theorem A.2}

\[ (S_{W})_{I} \subseteq (I_{W})_{T} \]

An abstract version of the "implementation" \( W \), thus if \( S_{W} \models I_{W} \) \( \models I_{W} \)

Since both and can be represented as an automaton, \( \models \) can be regarded as

- \( (\phi \models W \times \forall W)_{T} \)
- \( (\phi \models W \times \forall W)_{T} \)

Usually \( \phi \) is transformed to \( \phi \)-\( W \) and model checking consists of checking that

- Both components infinite often
- Component automata do - the infinite run must visit the reinitial states of
- The product automaton \( \phi \models W \times \forall W \) must accept an infinite word \( \phi \) both

\textbf{Models, automata and transition relations (3/3)}
(T \cup S) = \emptyset.

For any property \( T \) there exists a safety property \( \emptyset \) and a liveness property \( \emptyset \).

\( \emptyset \) is the only prop that is both safety and liveness.

Liveness props are closed under finite unions but not under intersections.

Safety props are closed under finite unions and arbitrary intersections.

Theorem 4.3 (Properties of safety and liveness properties)

\( \phi = \emptyset \lor \emptyset \lor \emptyset \)

A safety property if for all natural models \( \emptyset \).

\( \emptyset \) is a liveness property if the safety property is broken, there must be a finite prefix that can not be

\( \phi = \emptyset \lor \emptyset \lor \emptyset \lor \emptyset \)

\( \emptyset \) is a safety property if for all natural models \( \emptyset \).

\( \emptyset \) is a safety property if the safety property is broken, there must be a finite prefix that can not be

\( \emptyset \lor \emptyset \lor \emptyset \lor \emptyset \)

\( \emptyset \lor \emptyset \lor \emptyset \lor \emptyset \)

Safety and Liveness Properties (1/4)
Every temporal formula built from literals with $\land$, $\lor$, $\neg$, and $\diamond$ defines a safety property.

Theorem 4.4

A syntactical definition of an $\mathcal{LTL}$ safety property:

Safety Properties (2/4)
transition system for $\text{LTL}$ safety properties

there exists a tableau procedure (section 7) for obtaining a deterministic

this property:

for every $\mu$-regular safety property there is a finite transition system defining

$\text{Theorem 4.6}$

(all finite strings) the finitary requirement prevents the following example that defines that defines the finitary transition system defies a safety property:

$\text{Theorem 4.5}$

finite non-determinism allowed

transition system $S$, $S_0$ is finitary if $S_0$ is finite and is image finite: "only

$\{ \forall \in (S, x) \cap \exists \in S \}$

a binary relation $\forall \subseteq \exists \times \exists$ is image finite if for any $x \in \exists$ the set

Safety Properties: characterization by automata in (3/4)
Variation of 4.2 that was for all \(m\)-regular properties

\[
\phi \models \forall \mathcal{W} \exists \mathcal{W} \models \forall \phi
\]

If \(\mathcal{W} \models \phi\) then \(\mathcal{W} \models \phi\), \(\mathcal{W} \models \phi\) are mutually transition systems, \(\mathcal{W} \models \phi\) if for all safety properties

**Theorem 4.7**

\(\mathcal{W} \models \phi\) for all safety properties to establish \(\mathcal{W} \models \phi\) if \(\mathcal{W} \models \phi\) is sufficient to check whether \(\mathcal{W} \models \phi\) implies \(\mathcal{W} \models \phi\).

be used in specification-based testing

this can be checked by executing \(\forall \mathcal{W} \models \phi\) and concurrently in lock-step (can

\(\phi\) can be checked by the language containment problem (and \(\forall \mathcal{W} \models \phi\) \iff \exists \mathcal{W} \models \phi\) and

can be checked that a model sequence Validates an \(m\)-regular safety property can be

Safety Properties (in practice) (4/4)
In general, it is usually better idea to combine states rather than delete them.

Temporal properties

genrated submodel is the model consisting of all reachable states preserves all

part of a bigger graph

\[ m = I^m \]

\[ \Omega \uparrow \mathcal{I} = I \mathcal{I} \]

\[ \Omega \subseteq I^\Omega \]

\[ m \in \text{submodel of a model } \mathcal{M} \text{ if } I \mathcal{M} \subseteq I \mathcal{M} \]

System

- It may be useful to formulate properties regarding the structure of the
- Language containment for large nondeterministic systems is hard

- Weaker preorders than the language inclusion are useful

Simulation Relations (1/6)
\[(\mathcal{H}) \mathcal{I} \in (n, n)\]

If for every \( n \in \mathcal{I} \) and \( \mathcal{I} \) and \( \mathcal{H} \) there exists at most one \( n' \in \mathcal{I} \),

- For deterministic models \( \mathcal{W} \Leftrightarrow \mathcal{W} \Leftrightarrow \mathcal{W} \Leftrightarrow \mathcal{W} \Leftrightarrow \mathcal{W} \)

- If \( \mathcal{W} = \mathcal{W} \) then \( \mathcal{W} \Leftrightarrow \mathcal{W} \Leftrightarrow \mathcal{W} \)

- If \( \mathcal{W} \Leftrightarrow \mathcal{W} \) then \( \mathcal{W} \Leftrightarrow \mathcal{W} \Leftrightarrow \mathcal{W} \)

- Other properties:
- Simulation is a preorder on class of all models (4/8)
- \( \mathcal{W} \) is an abstraction of \( \mathcal{W} \), less states but more behaviours.
- The same transition as \( \mathcal{W} \) - one state of \( \mathcal{W} \) can simulate several states.
- To have a simulation \( \mathcal{W} \Leftrightarrow \mathcal{W} \Leftrightarrow \mathcal{W} \Leftrightarrow \mathcal{W} \Leftrightarrow \mathcal{W} \),

\[ \begin{align*}
H &= (n, n) \text{ and } (\mathcal{H}) \mathcal{I} \in (a, n) \\
& \text{and for all } H \in (a, n) : a \cdot n \mathcal{W} \Leftrightarrow \mathcal{W}
\end{align*} \]

- \( \mathcal{W} \Leftrightarrow \mathcal{W} \Leftrightarrow \mathcal{W} \Leftrightarrow \mathcal{W} \Leftrightarrow \mathcal{W} \)

- Other properties:

Simulations relations (6)
like 4.2 and 4.6, this allows checking the modal box for all modal box

\[ \phi \models \mathcal{M}_1 \]

formulas \( \phi \mathcal{M}_2 \) if \( \mathcal{M}_2 \) is a Kripke-model. \( \mathcal{M}_2 \models \mathcal{M}_1 \) implies that for all modal box

Theorem 4.8

formulas

\[ \phi \models \text{Kripke box formulas for all modal box} \]

- if \( \phi \) are modal box formulas, then \( \phi \wedge \phi \) are modal box

- if \( \phi \) are modal box formulas, then \( \phi \vee \phi \) are modal box

\[ \top, \bot \]

- if \( \phi \) are modal box formulas, then \( \phi \wedge \phi \) are modal box

\[ \text{simultaneous relation} (\phi, \phi) \]

Theorem 4.9

If there is a simultaneous relation between the models, then the models have a

Simultaneous Relations: Preserved Properties (3/6)
Theorem 4.9

Since distinguished properties of one path is not possible

\( \text{ACTL} \) formulas describe properties that are valid in all paths of the model.

- Internal formulas are in the dual form.
- Internal formulas are in the form of a formula without the existential quantifier.

\( (\phi + \bigwedge \psi) \lor (\phi + \Upsilon \psi) \lor (\phi \land \psi) \lor (\phi \lor \psi) \lor \top \lor \bot \)

Simulation relations preserved properties (4/6)
Theorem 4.11

A finite case the converse of 4.8 holds.

Theorem 4.10

the example can not be distinguished by any modal formulas.
Intersection of all $uH$ is the largest simulation relation.

\[ uH + 1 = uH \]

Since this is a finite model, eventually $uH$ Simulation transition should end to some other pair in some other model. This simulation has a transition that the simulation model can match – thus, $uH + 1$ is the model to be the relation.

1. Place all pairs of states with matching properties into the first iteration of non-deterministic models.

2. For the next iteration, place a pair of $uH + 1$ if the model to be $uH$. For $uH + 1$, place a pair of $uH$ to $uH$. For $uH$, place all pairs of states with matching properties into the first iteration of deterministic models.

An algorithm for creating a simulation relation between two languages.

For deterministic finite automata, there are efficient algorithms for language.
$\mathcal{A} \cong \mathcal{N}$

and $(\mathcal{Y})^1_\mathcal{I} \ni (n, n')$ then there exists $n$ s.t.

$\mathcal{A} \cong \mathcal{N}$

and $(\mathcal{Y})^2_\mathcal{I} \ni (n, n')$ then there exists $n$ s.t.

$(d) \mathcal{I} \ni (d) \mathcal{I} \ni (n, n')$ then $n \ni m \ni 0$

Kripke models $\mathcal{M}, \mathcal{M}'$:

bisimulation is an equivalence relation between universes of two

equivalence of the generated languages

submodel ordering $\equiv$ induces isomorphism, sequence valuation $\equiv$ indices

$\mathcal{M} \subset \mathcal{M}'$

$\mathcal{M} \subset \mathcal{M}$ if $\mathcal{M} \approx \mathcal{M}$ and

$a$ preorder $\subset$ can induce an equivalence an equivalence is a symmetric preorder

Bisimulations (1/4)

(TL-79.298, Postgraduate Course in Digital Systems Science

29 October 2001)
the relation $\equiv$ is the elementary equivalence

$\phi \models_1 \mathcal{M}_1 \iff \models_2 \mathcal{M}_2 \iff \models_2 \mathcal{M}_2$ for all well-formed formulas of $\mathcal{L}$ that hold in $\mathcal{M}_1$. $\equiv$ is equivalence w.r.t. to logic $\mathcal{L}$

way around

if $\models_1 \mathcal{M}_1 \models_2 \mathcal{M}_2$ then $\mathcal{M}_1 \equiv \mathcal{M}_2$ and $\mathcal{M}_2 \equiv \mathcal{M}_1$, but not necessarily the other

- a model is bisimilar to its unfolding
- a model is bisimilar to its reachable part
- outputs are removed
- each model is bisimilar to one where duplicate states with same inputs and

Some properties of bisimilar models

(2/14)
are image finite, then $\mathcal{M}_1 \equiv_{\text{ML}} \mathcal{M}_2$.

Image finite models are modally equivalent iff they are bisimilar: if $\mathcal{M}_1 \equiv_{\text{ML}} \mathcal{M}_2$.

**Theorem 5.3**

Converse requires image finiteness:

Bisimilar models are modally equivalent: if $\mathcal{M}_1 \mathcal{M}_2$, then $\mathcal{M}_1 \equiv_{\text{ML}} \mathcal{M}_2$.

**Theorem 5.2**

Formulas are bisimulation invariant:

Formulas (compare to modal box preservation of simulations (4.10)) – modal bisimulation relations are precisely those equivalences that preserve all modal bisimulations: preserved properties (3.14).
are finite, then $\mathcal{W}_1 \equiv \mathcal{W}_2$ if $\mathcal{W}_1 \subseteq \mathcal{W}_2$ if they are bisimilar. If $\mathcal{W}_1 \subseteq \mathcal{W}_2$.

Finite models are monotonic $\mathcal{T}_L$-equivalent if they are bisimilar.

**Theorem 5.5**

Since modal logic is a sublanguage of $\mathcal{T}_L$.

Let $\phi \in \mathcal{T}_L$ be a monotonic $\mathcal{T}_L$-formula. Then $\mathcal{W} \models \phi$ if $\mathcal{W} \models \phi$.

Let $\mathcal{W}$ be a finite model $(\mathcal{I}, \mathcal{O}) \models \mathcal{W}$, and let $\phi$ be a monotonic. $\mathcal{W}$, $\mathcal{W}$, $\mathcal{W}$, $\mathcal{W}$.

**Theorem 5.4**

Results for more expressive logics.

By restricting the model to finite Kripke-models, it is possible have similar.

**Bisimilar:** preserved properties ($\equiv$)
capabilities

Logics with different expressiveness can have the same distinguishing

\[ \text{TTL and thus } \text{TTL} \]

can be distinguished by a \text{TTL} formula as well - \text{TTL} can be transferred to

If two finite models can be distinguished by a formula of logic \text{TTL}, then they

are finite, then \( \mathcal{M}_1 \Rightarrow \mathcal{M}_2 \) if \( \mathcal{M}_1 \equiv_{\text{TTL}} \mathcal{M}_2 \).

Finite models are monomorphic / \text{TTL}-equivalent if they are distinguishable: if \( \mathcal{M}_1 \equiv_{\text{TTL}} \mathcal{M}_2 \),

Theorem 5.6

It is possible to use weaker logics to distinguish between models:

Distinguish: distinguishing power (6/14)
and vice versa – or iff for all models it holds that $\mathcal{M}_1 \equiv \mathcal{M}_2$ iff $\mathcal{L}_1 \equiv \mathcal{L}_2$ have the same distinguishing power if $\mathcal{L}_1$ is at least as distinguishing as

second

and vice versa – for each formula in one logic there is an equivalent one in the other logic $\mathcal{L}_1 \equiv \mathcal{L}_2$ if and only if $\mathcal{L}_1 \equiv \mathcal{L}_2$ implies $\mathcal{L}_2 \equiv \mathcal{L}_1$ are equivalent w.r.t. $\mathcal{L}_1$ and $\mathcal{L}_2$ are equivalent w.r.t. $\mathcal{L}_1$.

$\mathcal{L}_1 \equiv \mathcal{L}_2$ if and only if any two models that are equivalent or equivalent w.r.t. $\mathcal{L}_1$ are equivalent or equivalent w.r.t. $\mathcal{L}_2$.

$\mathcal{L}_1 \equiv \mathcal{L}_2$ if and only if any formula $\phi \in \mathcal{L}_2$ s.t. for all models $\mathcal{M}$ there exists a formula $\psi \in \mathcal{L}_1$ s.t. for any formula $\psi \in \mathcal{L}_1$ and for all models $\mathcal{M}$ there exists a formula $\psi \in \mathcal{L}_2$ s.t. for any formula $\psi \in \mathcal{L}_2$ and for all models $\mathcal{M}$ there exists a formula $\phi \in \mathcal{L}_1$ s.t.

Bisimulations, expressiveness and distinguishing power (6/14)
If $L_1$ is at most as expressive as $L_2$, then it is at most as distinguishing. If $L_2$ have the same expressive power, then they have the same distinguishing power but not vice versa.

Theorem 5.7

Expressiveness is a finer equivalence relation than distinguishability.

Bisimulations: expressiveness and distinguishability (§14)
Theorem 5.9 (Expressive Completeness of ML)

Translated to ML

and model expressiveness - species which 1st order formulas can be
the reverse direction provides a connection between bisimultations, first order

translated to ML

expressive logics like ML.

multimodal logics are bisimulation invariant (5.7), but this holds for more

preserved under bisimulations

a logic T is bisimulation invariant if all well-formed formulas of T are

holds that \( \forall M \models \varphi \rightarrow \exists \psi \models \psi \)

any formula \( \varphi \) is preserved under bisimulations if for all models \( M \models \varphi \)

Bisimultations: Varsity for expressiveness (8/14)
\( \text{MOS} \) can be translated to \( \text{ TT} \) every logic which is bisimulation invariant and has a semantic translation to
definable by positive \( \text{TT} \) formulas.

Let \( \phi \) be any \( \text{MOS} \) property. Then \( \phi \) is preserved under bisimulations iff

\textbf{Theorem 5.10 (Expressive completeness of \( \text{ TT} \))}

the same result can be extended to 2nd order formulas and

\( \text{Bisimulations: Yaroslav} \text{ for expressiveness (9/14)} \)
wins

- If Bob cannot match Ann's move he loses; if he can play the game forever he
  that board and placing the piece hounoring the transition relation
  Bob has to match Ann's move on the other board by locating the ith piece on
  relation w.r.t. placed pieces
  Ann places her (i + 1)th piece on either of the boards hounoring the transition
  different moves - labels must match or Bob loses
  game is played on two Kripke-structures, both place their first pieces on
  pieces: q0, q1, . . . and p0, p1, . . .
  two players: Ann and Bob, each having an unlimited number of identical
  (logic)
  a convenient way of imaging bisimulations (and equivalences w.r.t. other

Bisimulations: Ehrnfeucht-Fraisse Games (10/14)
Theorem 5.11

The winning strategy if \( n \) wins does not

immediately or after \( n - 1 \) rounds

\[ \frac{2}{3} / 14 \]

Significantly: Ehrlich-Feinstein Games (10/14)

Different solutions by MIS\( \cdot \)O

Allowing sets of pieces \( \text{Ann has a winning strategy if the boards can be} \)

winning strategy if they are dissimilar.

Ann has a winning strategy if the two models are not dissimilar; Bob has a
For each auto-bisimulation the greatest equivalence relation \( \equiv \) that

the model.

Thus the greatest auto-bisimulation is the union of all auto-bisimulations in

\[ \text{Theorem 5.12} \]

model: auto-bisimulations

\[ \text{note that all definitions have not forbidden bisimulations to points in the same} \]

\[ \text{to minimize a Knapsack model w.r.t. bisimulations} \]

\[ \text{Bisimulations: auto-bisimulations (11/14)} \]
The quotient of the model w.r.t. its largest auto-bisimulation is the minimal \( \mathcal{W} \equiv \mathcal{W} \).

For any model of \( \mathcal{W} \) and equivalence relation \( \equiv \), \( \mathcal{W} \) for auto-bisimulations (12/14) → (12/14).
The coarsest stable partition is the largest auto-bisimulaiton.

**Theorem 5.14**

A partition is stable if all components are uniform and stable w.r.t. other

It is possible to access from partition

\[
\{\} = d < \mathcal{U} > \cup_{\mathcal{U} \in m} \wedge d < \mathcal{U} > \mathcal{V} \equiv m : \mathcal{U}
\]

A component \( m \in \mathcal{U} \) is stable w.r.t. set \( \mathcal{U} \)

The nodes in partition have the same labeling (proposition).

\[
\{\} = (d)\mathcal{I} \cup_{\mathcal{I} \in m} \wedge (d)\mathcal{I} \mathcal{V} \equiv m : d \in d\mathcal{A}
\]

Given a partition to equivalence classes, a component \( m \in \equiv \mathcal{U} \) is uniform if

\[
\text{those nodes that have a transition to } d
\]

\[
\{ (\mathcal{U}) \in (\mathcal{I} \in m) : d \in \mathcal{I} \mathcal{E} \in m \} = d < \mathcal{U} > \cup \mathcal{V} \equiv m
\]

For any set of points \( d \) partitions \((\mathcal{I} \in m) \cup (\mathcal{V} \equiv m) \)

**Bisimulation partitions (13/14)**
Bisimulation: Algorithm (14/14)