Computing Minimal Models Modulo Subset-Simulation for Modal Logics

Fabio Papacchini   Renate A. Schmidt

School of Computer Science
The University of Manchester

September 20, 2013

(Minimal) Model Generation

Useful for several tasks:

- hardware and software verification
- fault analysis
- commonsense reasoning
- ...

They have been investigated for many logics.

Minimality Criteria

Several minimality criteria has already been considered:

- domain minimality
- minimisation of a certain set of predicates
- minimal Herbrand models

Minimality Criteria

Several minimality criteria has already been considered:

- domain minimality
- minimisation of a certain set of predicates
- minimal Herbrand models

Aims

To propose a new minimality criterion for modal logics that

- takes in consideration the semantics of models
- is generic enough to be applied to a variety of modal logics

To propose a tableau calculus for the generation of these minimal models

Modal Logics

Syntax

$$\phi = \top \mid \bot \mid p_i \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \langle R_i \rangle \phi \mid [R_i]\phi \mid \langle \mathcal{U} \rangle \phi \mid [\mathcal{U}]\phi$$

Semantics, $M = (W, \{R_1, \ldots, R_n\}, V)$

$M, u \not\models \bot$	$M, u \models \top$
$M, u \models p_i$	iff $p_i \in V(u)$
$M, u \models \neg\phi$	iff $M, u \not\models \phi$
$M, u \models \phi_1 \vee \phi_2$	iff $M, u \models \phi_1$ or $M, u \models \phi_2$
$M, u \models \phi_1 \wedge \phi_2$	iff $M, u \models \phi_1$ and $M, u \models \phi_2$
$M, u \models [R_i]\phi$	iff for every $v \in W$ if $(u, v) \in R_i$ then $M, v \models \phi$
$M, u \models \langle R_i \rangle \phi$	iff there is a $v \in W$ such that $(u, v) \in R_i$ and $M, v \models \phi$
$M, u \models [\mathcal{U}]\phi$	iff for every $v \in W$ $M, v \models \phi$
$M, u \models \langle \mathcal{U} \rangle \phi$	iff there is a $v \in W$ such that $M, v \models \phi$

Why a New Minimality Criterion?

Domain minimal models

Advantages:

- models with the smallest domain
- finite models for logics with the finite model property

Disadvantages:

- models can be counter-intuitive
- hard to achieve minimal model completeness

Why a New Minimality Criterion?

Domain minimal models

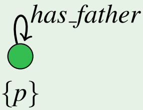
Advantages:

- models with the smallest domain
- finite models for logics with the finite model property

Disadvantages:

- models can be counter-intuitive
- hard to achieve minimal model completeness

$$\langle has\_father\rangle p$$



Why a New Minimality Criterion? (cont'd)

Minimal Herbrand models

Advantages:

- minimisation of relations and atoms
- comparison of atoms between the same world in different models

Disadvantages:

- the criterion is syntactic
- minimal models can be infinite

Why a New Minimality Criterion? (cont'd)

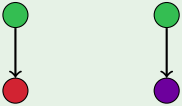
Minimal Herbrand models

Advantages:

- minimisation of relations and atoms
- comparison of atoms between the same world in different models

Disadvantages:

- the criterion is syntactic
- minimal models can be infinite



Why a New Minimality Criterion? (cont'd)

Minimal Herbrand models

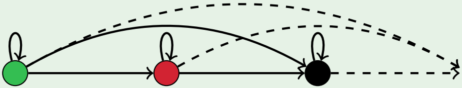
Advantages:

- minimisation of relations and atoms
- comparison of atoms between the same world in different models

Disadvantages:

- the criterion is syntactic
- minimal models can be infinite

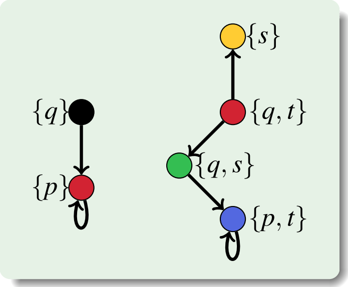
$\Box\Diamond\top$ in a transitive and reflexive frame



Subset-Simulation Relation $S_\subseteq$

Relation between nodes of two models $M = (W, \{R_1, \ldots, R_n\}, V)$ and $M' = (W', \{R_1, \ldots, R_n\}, V')$ s.t.

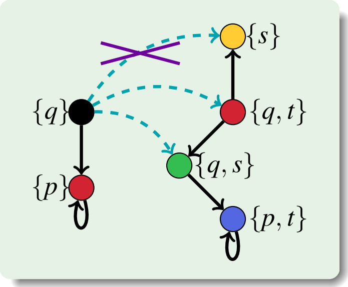
1. the subset relationship holds ($V(u) \subseteq V'(u')$)
2. successor in the first model
   $\Rightarrow$ successor in the second model
3. 1 and 2 hold for the successors of point 2



Subset-Simulation Relation $S_{\subseteq}$

Relation between nodes of two models $M = (W, \{R_1, \ldots, R_n\}, V)$ and $M' = (W', \{R_1, \ldots, R_n\}, V')$ s.t.

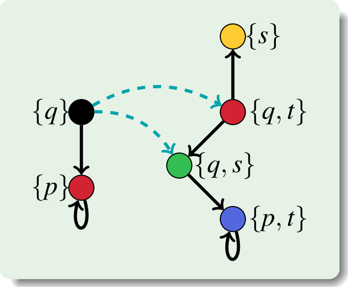
1. the subset relationship holds ($V(u) \subseteq V'(u')$)
2. successor in the first model $\Rightarrow$ successor in the second model
3. 1 and 2 hold for the successors of point 2



Subset-Simulation Relation $S_{\subseteq}$

Relation between nodes of two models $M = (W, \{R_1, \ldots, R_n\}, V)$ and $M' = (W', \{R_1, \ldots, R_n\}, V')$ s.t.

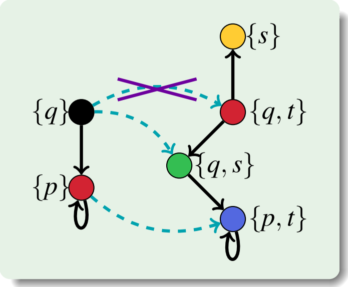
1. the subset relationship holds ($V(u) \subseteq V'(u')$)
2. successor in the first model
   $\Rightarrow$ successor in the second model
3. 1 and 2 hold for the successors of point 2



Subset-Simulation Relation $S_{\subseteq}$

Relation between nodes of two models $M = (W, \{R_1, \ldots, R_n\}, V)$ and $M' = (W', \{R_1, \ldots, R_n\}, V')$ s.t.

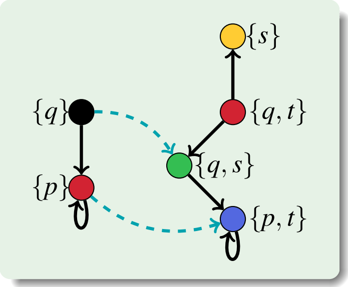
1. the subset relationship holds ($V(u) \subseteq V'(u')$)
2. successor in the first model
   $\Rightarrow$ successor in the second model
3. 1 and 2 hold for the successors of point 2



Subset-Simulation Relation $S_{\subseteq}$

Relation between nodes of two models $M = (W, \{R_1, \ldots, R_n\}, V)$ and $M' = (W', \{R_1, \ldots, R_n\}, V')$ s.t.

1. the subset relationship holds ($V(u) \subseteq V'(u')$)
2. successor in the first model
   $\Rightarrow$ successor in the second model
3. 1 and 2 hold for the successors of point 2



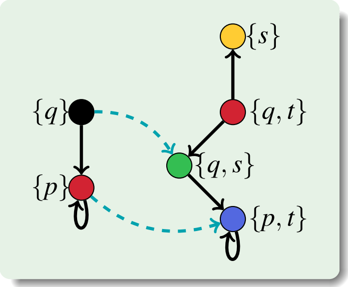
Full Subset-Simulation: for all $u \in W$ there exists some $u' \in W'$ s.t. $uS_{\subseteq}u'$.

Maximal Subset-Simulation: $S_{\subseteq}$ maximal if there is no $S'_{\subseteq}$ s.t. $S_{\subseteq} \subset S'_{\subseteq}$.

Subset-Simulation Relation $S_{\subseteq}$

Relation between nodes of two models $M = (W, \{R_1, \ldots, R_n\}, V)$ and $M' = (W', \{R_1, \ldots, R_n\}, V')$ s.t.

1. the subset relationship holds ($V(u) \subseteq V'(u')$)
2. successor in the first model
   $\Rightarrow$ successor in the second model
3. 1 and 2 hold for the successors of point 2



Full Subset-Simulation: for all $u \in W$ there exists some $u' \in W'$ s.t. $uS_{\subseteq}u'$.

Maximal Subset-Simulation: $S_{\subseteq}$ maximal if there is no $S'_{\subseteq}$ s.t. $S_{\subseteq} \subset S'_{\subseteq}$.

If there is a full and maximal subset-simulation from M to M', then M is subset-simulated by M', or M' subset-simulates M.

Models Minimal Modulo Subset-Simulation

Subset-simulation is

- reflexive
- transitive

Models Minimal Modulo Subset-Simulation

Subset-simulation is

- reflexive
- transitive

$\Longrightarrow$ a preorder

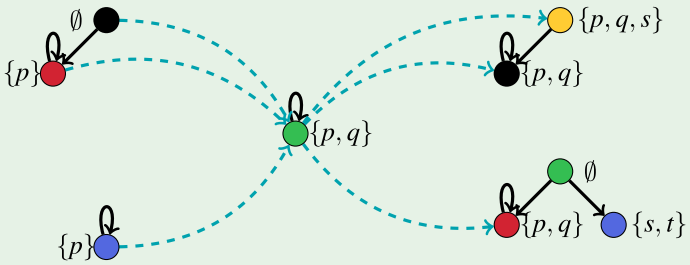
Models Minimal Modulo Subset-Simulation

Subset-simulation is

- reflexive
- transitive

$\Longrightarrow$ a preorder

Minimal models are the minimal elements of the preorder.



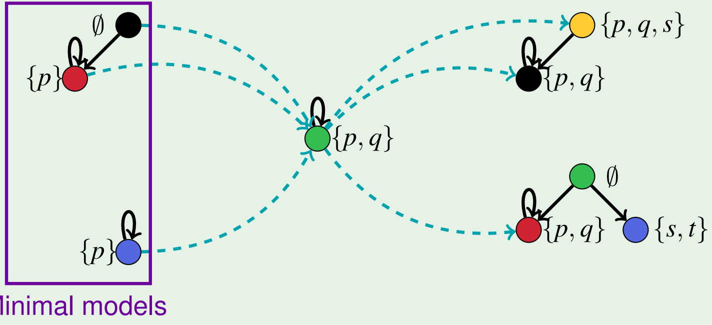
Models Minimal Modulo Subset-Simulation

Subset-simulation is

- reflexive
- transitive

$\Longrightarrow$ a preorder

Minimal models are the minimal elements of the preorder.



Too Many Minimal Models! – Symmetry Classes

As subset-simulation is not a partial order

- there exist symmetry classes of minimal models
- symmetric minimal models are not equivalent
- a symmetry class can have infinitely many minimal models

Too Many Minimal Models! – Symmetry Classes

As subset-simulation is not a partial order

- there exist symmetry classes of minimal models
- symmetric minimal models are not equivalent
- a symmetry class can have infinitely many minimal models

How can we make the minimality criterion stricter?

Refining Symmetric Models – Simulation

Simulation is as subset-simulation except for the condition $V(u) = V'(u')$.

The use of simulation among symmetric minimal models allows to

- reduce the number of minimal models
- recognise bisimilar models

Symmetric w.r.t. subset-simulation:

The right model is simulated by the left model, but not the other way around:

Properties of the Minimality Criterion

- applied to the graph representation of models (syntax independent)
- loop free models are preferred
- minimisation of the content of worlds
- suitable for many non-classical logics

Tableau Calculus

Input: a modal formula in negation normal form.

Tableau Calculus

Input: a modal formula in negation normal form.

Selection-based resolution:

- closure rule
- removes negative information from disjunctions

$$(SBR) \quad \frac{u : p_1 \;\ldots\; u : p_n \qquad u : \neg p_1 \vee \ldots \vee \neg p_n \vee \Phi_\alpha^+}{u : \Phi_\alpha^+}$$

Φ_α^+: a disjunction where no disjunct is of the form $\neg p_i$.

Tableau Calculus

Input: a modal formula in negation normal form.

Selection-based resolution:

- closure rule
- removes negative information from disjunctions

$$(SBR)\ \frac{u : p_1 \ \ \ldots \ \ u : p_n \qquad u : \neg p_1 \vee \ldots \vee \neg p_n \vee \Phi_\alpha^+}{u : \Phi_\alpha^+}$$

Lazy clausification:

- avoids preprocessing steps
- can result in less inferences

$$(\alpha)\ \frac{u : (\phi_1 \wedge \ldots \wedge \phi_n) \vee \Phi_\alpha^+}{\begin{array}{c} u : \phi_1 \vee \Phi_\alpha^+ \\ \vdots \\ u : \phi_n \vee \Phi_\alpha^+ \end{array}}$$

Φ_α^+: a disjunction where no disjunct is of the form $\neg p_i$.

Tableau Calculus (cont'd)

Complement splitting:

- variation of the standard β rule
- detects trivially non-minimal models

$$(\beta)\ \frac{u : \mathcal{A} \vee \Phi^+}{\begin{array}{l|l} u : \mathcal{A} & u : \Phi^+ \\ u : neg(\Phi^+) & \end{array}}$$

$\mathcal{A} ::= p \mid \langle R_i \rangle \phi \mid [R_i]\phi$
$neg(\Phi^+) = \neg p_1 \wedge \ldots \wedge \neg p_n$

Φ^+: a disjunction where no disjunct is of the form $\neg p_i$ or is a conjunction.

Tableau Calculus (cont'd)

Complement splitting:

- variation of the standard β rule
- detects trivially non-minimal models

$$(\beta)\ \frac{u : \mathcal{A} \vee \Phi^+}{\begin{array}{l|l} u : \mathcal{A} & u : \Phi^+ \\ u : neg(\Phi^+) & \end{array}}$$

$\mathcal{A} ::= p \mid \langle R_i \rangle \phi \mid [R_i]\phi$
$neg(\Phi^+) = \neg p_1 \wedge \ldots \wedge \neg p_n$

Expansion of diamond formulae:

$$(\Diamond)\ \frac{u : \langle R_i \rangle \phi}{\begin{array}{l|l|l|l} (u, u_1) : R_i & \ldots & (u, u_n) : R_i & (u, v) : R_i \\ u_1 : \phi & & u_n : \phi & v : \phi \end{array}}$$

v is a fresh new world

Φ^+: a disjunction where no disjunct is of the form $\neg p_i$ or is a conjunction.

Tableau Calculus (cont'd)

Complement splitting:

- variation of the standard β rule
- detects trivially non-minimal models

$$(\beta)\ \frac{u : \mathcal{A} \vee \Phi^+}{\begin{array}{l|l} u : \mathcal{A} & u : \Phi^+ \\ u : neg(\Phi^+) & \end{array}}$$

$$\mathcal{A} ::= p \mid \langle R_i \rangle \phi \mid [R_i]\phi$$
$$neg(\Phi^+) = \neg p_1 \wedge \ldots \wedge \neg p_n$$

Expansion of diamond formulae:

$$(\Diamond)\ \frac{u : \langle R_i \rangle \phi}{\begin{array}{l|l|l|l} (u, u_1) : R_i & \ldots & (u, u_n) : R_i & (u, v) : R_i \\ u_1 : \phi & & u_n : \phi & v : \phi \end{array}}$$

v is a fresh new world

Expansion of box formulae: the standard $\Box$ rule

Φ^+: a disjunction where no disjunct is of the form $\neg p_i$ or is a conjunction.

Properties of the Tableau Calculus

The calculus is

- refutationally sound and complete
- minimal model complete (generates all minimal models)

Properties of the Tableau Calculus

The calculus is

- refutationally sound and complete
- minimal model complete (generates all minimal models)

But it is not minimal model sound (generates also non-minimal models)!

Minimal Model Soundness

Idea: incremental generation of models

Expansion strategy: the left most branch with the least number of worlds

Subset-simulation test:

- early closure of “non-minimal” branches
- backward closure of branches - minimal model refining

Minimal Model Soundness

Idea: incremental generation of models

Expansion strategy: the left most branch with the least number of worlds

Subset-simulation test:

- early closure of “non-minimal” branches
- backward closure of branches - minimal model refining

The resulting calculus is minimal model sound and complete
$\Rightarrow$ all and only minimal models are generated.

Subset-Simulation Test

Early closure of "non-minimal" branches

A partial model M subset-simulates an extracted model M', but not the other way around.

- M is already not minimal
- no expansion of M can be minimal

  $\Rightarrow$ close the branch from which M is extracted

Subset-Simulation Test (cont'd)

Backward closure of branches - minimal model refining

$M =$ newly extracted model, $S =$ current set of minimal models.

Compare M with all $M' \in S$, close branches accordingly and refine S.

- M is not minimal
  - close the branch from which M was extracted
- for all $M' \in S$ s.t. M' subset-simulates M, but no the other way around
  - remove all M' from S
  - close the branches from which all M' were extracted
  - add M to S
- for all $M' \in S$ s.t. M' subset-simulates M, and M subset-simulates M'
  - check for simulation

Subset-Simulation Test (cont'd)

Backward closure of branches - minimal model refining

$M =$ newly extracted model, $S =$ current set of minimal models.

Compare M with all $M' \in S$, close branches accordingly and refine S.

- M is not minimal
  - close the branch from which M was extracted
- for all $M' \in S$ s.t. M' subset-simulates M, but no the other way around
  - remove all M' from S
  - close the branches from which all M' were extracted
  - add M to S
- for all $M' \in S$ s.t. M' subset-simulates M, and M subset-simulates M'
  - check for simulation

Subset-Simulation Test (cont'd)

Backward closure of branches - minimal model refining

$M =$ newly extracted model, $S =$ current set of minimal models.

Compare M with all $M' \in S$, close branches accordingly and refine S.

- M is not minimal
  - close the branch from which M was extracted
- for all $M' \in S$ s.t. M' subset-simulates M, but no the other way around
  - remove all M' from S
  - close the branches from which all M' were extracted
  - add M to S
- for all $M' \in S$ s.t. M' subset-simulates M, and M subset-simulates M'
  - check for simulation

Subset-Simulation Test (cont'd)

Backward closure of branches - minimal model refining

$M =$ newly extracted model, $S =$ current set of minimal models.

Compare M with all $M' \in S$, close branches accordingly and refine S.

- M is not minimal
  - close the branch from which M was extracted
- for all $M' \in S$ s.t. M' subset-simulates M, but no the other way around
  - remove all M' from S
  - close the branches from which all M' were extracted
  - add M to S
- for all $M' \in S$ s.t. M' subset-simulates M, and M subset-simulates M'
  - check for simulation

Extending the Calculus

Structural rules for frame properties (reflexivity, transitivity, . . .)

$$(\mathbf{4})\ \frac{(u,v) : R_i \qquad\qquad (v,w) : R_i}{(u,w) : R_i}$$

Rules for universal modalities ($\langle\mathcal{U}\rangle$ and $[\mathcal{U}]$)

$$(\langle\mathcal{U}\rangle)\ \frac{u : \langle\mathcal{U}\rangle\phi}{u_1 : \phi \mid \ldots \mid u_n : \phi \mid v : \phi}$$

v is a fresh new world

Extending the Calculus

Structural rules for frame properties (reflexivity, transitivity, . . .)

$$(\mathbf{4})\ \frac{(u,v) : R_i \qquad (v,w) : R_i}{(u,w) : R_i}$$

Rules for universal modalities ($\langle\mathcal{U}\rangle$ and $[\mathcal{U}]$)

$$(\langle\mathcal{U}\rangle)\ \frac{u : \langle\mathcal{U}\rangle\phi}{u_1 : \phi \mid \ldots \mid u_n : \phi \mid v : \phi}$$

v is a fresh new world

Those extensions preserve minimal model soundness and completeness.
Termination depends on the extension (logic expressiveness).

Conclusion and Further Work

- minimality modulo subset-simualtion is
  - semantic (based on the graph representation)
  - suitable for many non-classical logics
- the tableau calculus
  - is minimal model sound and complete
  - can be generalised to cover more expressive logics
  - does not terminate for all the logics

Conclusion and Further Work

- minimality modulo subset-simualtion is
  - semantic (based on the graph representation)
  - suitable for many non-classical logics
- the tableau calculus
  - is minimal model sound and complete
  - can be generalised to cover more expressive logics
  - does not terminate for all the logics

- efficient implementation of the calculus
- study of reasonable restrictions for reducing the search space
  - how to simplify the $(\Diamond)$ rule?
  - how to achieve termination for logics with the finite model property?
- generalise the minimality criterion to fragments of first-order logic