

Ranked Predicate Abstraction for Branching Time: Complete, Incremental, and Precise

Harald Fecher¹ Michael Huth²

¹Christian-Albrechts-University at Kiel, Germany

²Imperial College London, United Kingdom

Beijing, ATVA 2006

Main Issues

Foundation for counter-example-guided abstraction refinement (CEGAR) for the **full** μ -calculus:

Development of extended predicate abstraction:

- sound,
- precise,
- incremental, and
- complete

Introduction

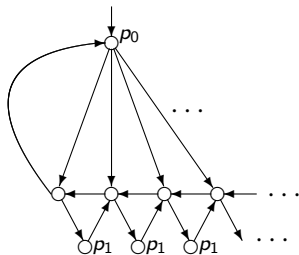
- Branching time (multiple system observers; biological systems)
- Branching time logic: **mu-calculus** having least and greatest fixpoints
- Model checking not directly applicable on **large** or **infinite** systems
- Counter-example-guided abstraction refinement (CEGAR):
initial abstraction; model check; spurious counterexample \rightarrow
refinement; loop
- Abstraction technique: **predicate abstraction**
(synthesized automatically using theorem prover)

Predicate abstraction

Divide concrete state space by a set of predicates: abstract state is subset of predicates (related concrete are those satisfying the contained predicates and not satisfying the omitted).

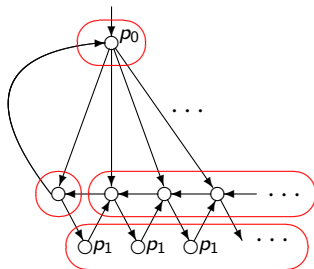
Mu-calculus needs over approximation (**may**-transition) and under approximation (**must**-transition). Must-hypertransition increase expressiveness.

Predicate abstraction illustration



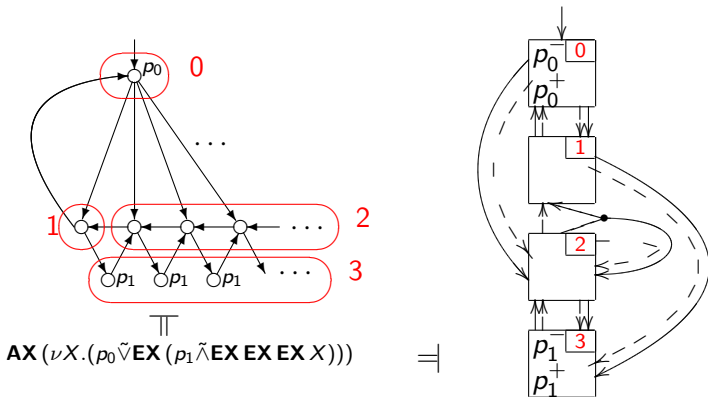
$$\text{AX} (\nu X. (\overset{\top}{p_0} \tilde{\vee} \text{EX} (p_1 \tilde{\wedge} \text{EX EX EX X})))$$

Predicate abstraction illustration



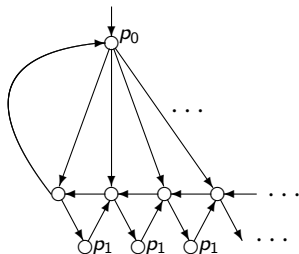
$$\mathbb{T} \\ \mathbf{AX} (\nu X. (p_0 \tilde{\vee} \mathbf{EX} (p_1 \tilde{\wedge} \mathbf{EX} \mathbf{EX} \mathbf{EX} X)))$$

Predicate abstraction illustration



Current predicate abstraction insufficient

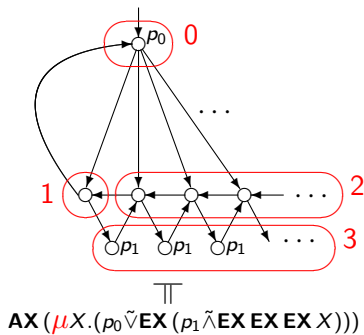
Problem: least fixpoint formulas



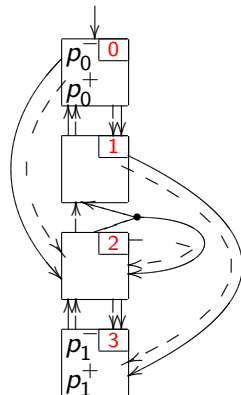
$$\text{AX} (\mu X. (p_0 \tilde{\vee} \text{EX} (p_1 \tilde{\wedge} \text{EX EX EX X})))$$

Current predicate abstraction insufficient

Problem: least fixpoint formulas

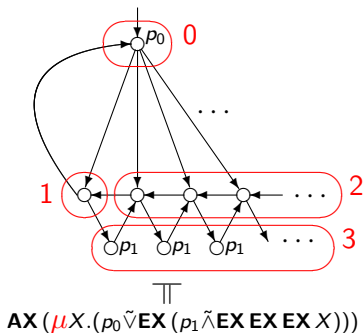


\neq

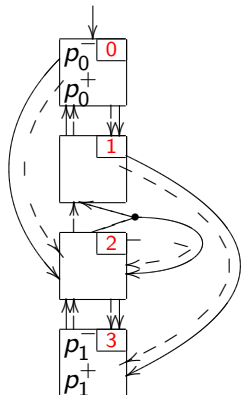


Current predicate abstraction insufficient

Problem: least fixpoint formulas



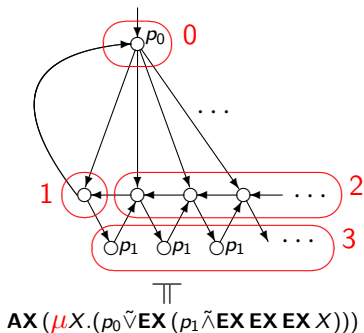
\neq



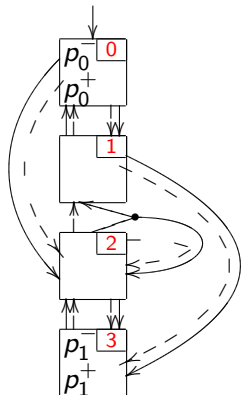
No other predicate abstraction does.

Current predicate abstraction insufficient

Problem: least fixpoint formulas



\neq



No other predicate abstraction does.

Solution: **ranking functions**

Ranked predicate abstraction

Definition

A *ranked predicate abstraction* \aleph of a state space S is a tuple (I, h) where

- $h: S \rightarrow I$ is a surjective function mapping concrete (S) to abstract (I) states

Ranked predicate abstraction

Definition

A *ranked predicate abstraction* \aleph of a state space S is a tuple $(I, h, (\leq^k)_{k \in K})$ where

- $h: S \rightarrow I$ is a surjective function mapping concrete (S) to abstract (I) states
- for all $k \in K$, with K a (possibly empty) index set, $\leq^k \subseteq (S \rightarrow \mathcal{P}(S)) \times (S \rightarrow \mathcal{P}(S))$ is a pre-order with well-founded irreflexive version $<^k$;

Ranked predicate abstraction

Definition

A *ranked predicate abstraction* \aleph of a state space S is a tuple $(I, h, J, (\leq^k)_{k \in K})$ where

- $h: S \rightarrow I$ is a surjective function mapping concrete (S) to abstract (I) states
- J is a non-empty set of rank locations;
[think J to be the subproperties]
- for all $k \in K$, with K a (possible empty) index set,
 $\leq^k \subseteq (S \times J) \times (S \times J)$ is a pre-order with well-founded irreflexive version $<^k$;

Ranked predicate abstraction

Definition

A *ranked predicate abstraction* \aleph of a state space S is a tuple $(I, h, J, (\leq^k)_{k \in K})$ where

- $h: S \rightarrow I$ is a surjective function mapping concrete (S) to abstract (I) states
- J is a non-empty set of rank locations;
[think J to be the subproperties]
- for all $k \in K$, with K a (possible empty) index set,
 $\leq^k \subseteq (S \times J) \times (S \times J)$ is a pre-order with well-founded irreflexive version $<^k$;
- $|I| + |J| + |K|$ is finite.

Hypermixed Kripke structures

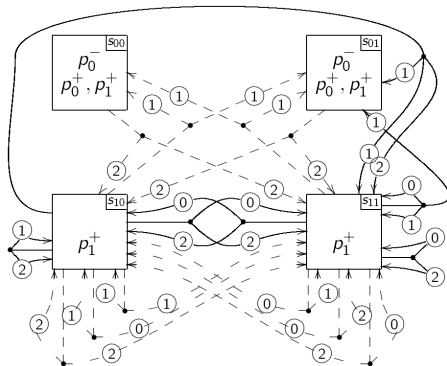
The abstract model has to be extended by

- Fairness constraints (Streett over transitions naturally occur) and
- May-hypertransition (conjunctively interpreted) for handling J .

Hypermixed Kripke structures

The abstract model has to be extended by

- Fairness constraints (Streitt over transitions naturally occur) and
- May-hypertransition (conjunctively interpreted) for handling J .

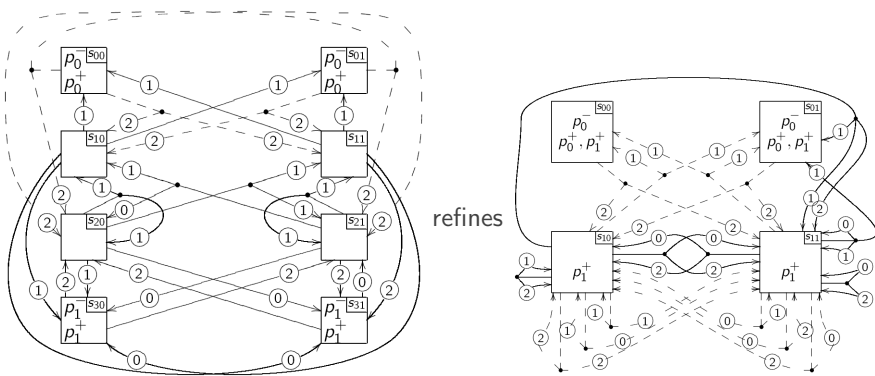


Streitt: Infinite 1-transitions \Rightarrow infinite 2-transitions

Hypermixed Kripke structures

The abstract model has to be extended by

- Fairness constraints (Streett over transitions naturally occur) and
- May-hypertransition (conjunctively interpreted) for handling J .



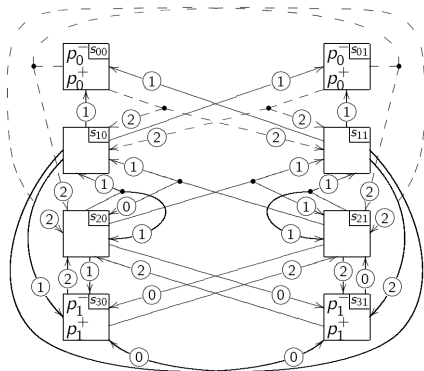
Streett: Infinite 1-transitions \Rightarrow infinite 2-transitions

Satisfaction

Via Games:

- in **EX**: Verifier choose must hypertrans;
Refuter choose element from target
- in **AX**: Refuter choose may hypertrans;
Verifier choose element from target
- Verifier wins infinite plays:
Non-acceptance at the model or acceptance at the property

Satisfaction example



$$S_{00} \models \mathbf{AX} (\mu X. (p_0 \checkmark \mathbf{EX} (p_1 \checkmark \mathbf{EX} \mathbf{EX} \mathbf{EX} X)))$$

AX: Player I chooses s_{10}^2 or s_{20}^2

EX-circle: Player I chooses must-transition to $\{s_{31}^0\}$ — she chooses must-transition to $\{s_{21}^0\}$ — she chooses must-transition to $\{s_{10}^1, s_{20}^1\}$ — she chooses must-transition to s_{01}^1 , resp. to $\{s_{10}^1, s_{20}^1\}$

\Rightarrow either p_0 is reached or non-acceptant model sequence

Soundness

Winning conditions for satisfaction are Rabin conditions (since Streett \Rightarrow RabinChain).

Thus deciding satisfaction is in NP

Theorem (Soundness)

Suppose M_1 refines M_2 and ϕ is mu-calculus formula:

$$M_2 \models \phi \Rightarrow M_1 \models \phi$$

\aleph -abstraction game

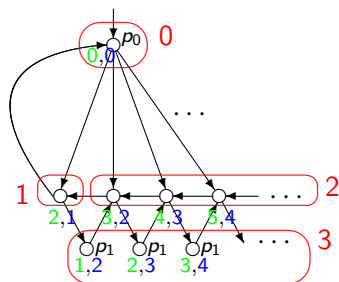
Player I tries to show that model M_1 is abstracted by model M_2 up to ranked predicate abstraction \aleph (is \aleph -abstracted by):

Player II can additionally **switch** between states of M_1 that map to the same elements via the abstraction function h as long as **no contradiction** to the ranking functions of \aleph is produced. Player I controls the ranking positions J .

Theorem

If M_1 is \aleph -abstracted by M_2 , then M_1 is abstracted by M_2 .

Precise abstraction



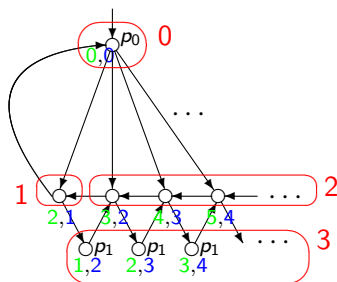
$J = \{g, b\}$ and $(s', j') \leq^0 (s, j) \Leftrightarrow \omega(s', j') \leq \omega(s, j)$

where $\omega(s, j)$ is depicted with colors

Precise abstraction

State space: $I \times J \times (K \rightarrow \{0, 1, 2\})$

function indicates for $k \in K$ if \leq^k remains equal, decrease, or increase



$J = \{g, b\}$ and $(s', j') \leq^0 (s, j) \Leftrightarrow \omega(s', j') \leq \omega(s, j)$
 where $\omega(s, j)$ is depicted with colors

$$\begin{array}{c} p_0^-^{s_{00}} \\ p_0^+ \end{array}$$

$$\begin{array}{c} p_0^-^{s_{01}} \\ p_0^+ \end{array}$$

$$\begin{array}{c} s_{10} \end{array}$$

$$\begin{array}{c} s_{11} \end{array}$$

$$\begin{array}{c} s_{20} \end{array}$$

$$\begin{array}{c} s_{21} \end{array}$$

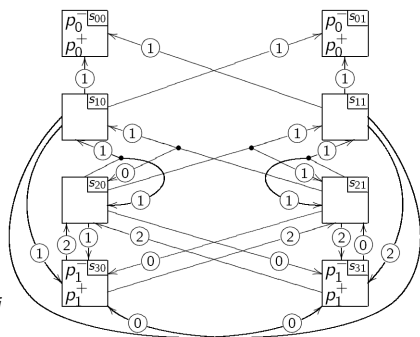
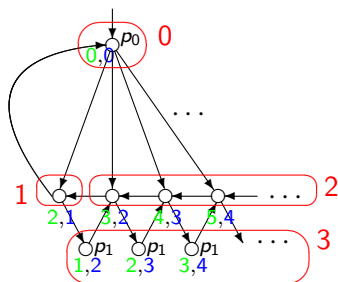
$$\begin{array}{c} p_1^-^{s_{30}} \\ p_1^+ \end{array}$$

$$\begin{array}{c} p_1^-^{s_{31}} \\ p_1^+ \end{array}$$

Precise abstraction

State space: $I \times J \times (K \rightarrow \{0, 1, 2\})$

function indicates for $k \in K$ if \leq^k remains equal, decrease, or increase

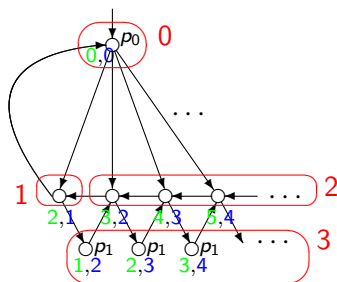


$J = \{g, b\}$ and $(s', j') \leq^0 (s, j) \Leftrightarrow \omega(s', j') \leq \omega(s, j)$
 where $\omega(s, j)$ is depicted with colors

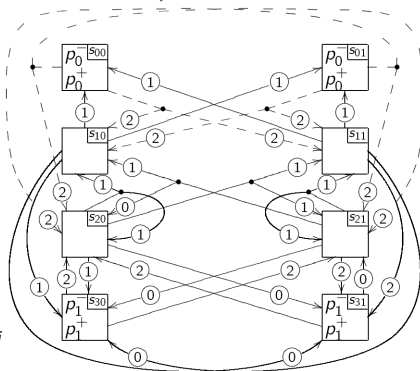
Precise abstraction

State space: $I \times J \times (K \rightarrow \{0, 1, 2\})$

function indicates for $k \in K$ if \leq^k remains equal, decrease, or increase



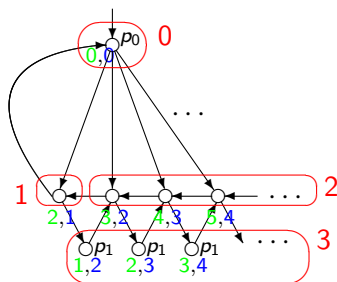
$J = \{g, b\}$ and $(s', j') \leq^0 (s, j) \Leftrightarrow \omega(s', j') \leq \omega(s, j)$
 where $\omega(s, j)$ is depicted with colors



Precise abstraction

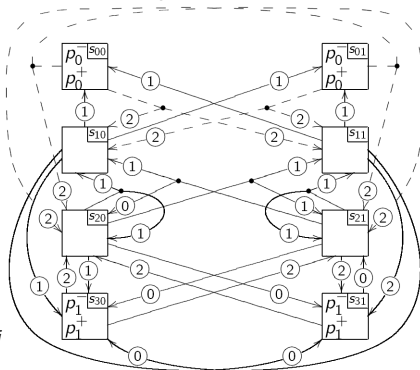
State space: $I \times J \times (K \rightarrow \{0, 1, 2\})$

function indicates for $k \in K$ if \leq^k remains equal, decrease, or increase



$J = \{g, b\}$ and $(s', j') \leq^0 (s, j) \Leftrightarrow \omega(s', j') \leq \omega(s, j)$

where $\omega(s, j)$ is depicted with colors



Streett fairness: at any $k \in K$, if the state function (third component) at k is infinitely often 1, then it is also infinitely often 2.

Preciseness

Theorem (Precision)

The defined abstraction M_{\aleph} is finite and a precise \aleph -abstraction, i.e.,

- *M_{\aleph} is a \aleph -abstraction of M and*
- *if M_2 is a \aleph -abstraction of M , then M_2 abstracts M_{\aleph} .*

Incremental

Definition

\aleph_1 is an *extension* of \aleph_2 if the partition is finer and only ranking functions are added.

Theorem

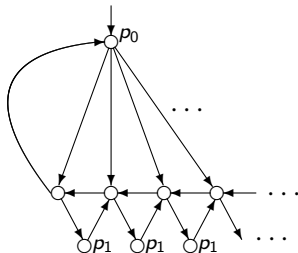
If \aleph_1 is an extension of \aleph_2 , then M_{\aleph_1} is abstracted by M_{\aleph_2} .

Theorem (Confluence of extensions)

For \aleph_1 and \aleph_2 there is constructible predicate abstraction being an extension of \aleph_1 and of \aleph_2 .

Non-trivial ranking positions J necessary for completeness

There is no ranked predicate abstraction \aleph of



such that its J is a singleton and its abstraction satisfies $\mathbf{AX}(\mu X.(p_0 \check{\vee} \mathbf{EX}(p_1 \check{\wedge} \mathbf{EX} \mathbf{EX} \mathbf{EX} X)))$.

We already saw that it is possible with non-singleton J .

Completeness

Let M Kripke structure and θ memoryless strategy for $M \models \phi$.

Partition (function h_θ): states are equivalent if they satisfy same subformulas via θ and θ behaves same on $\tilde{\forall}$ -properties

Ranking locations J : set of subproperties

Ranking function $\omega^{\theta,k}$: the least number of unfoldings necessary to guarantee that no further $2k + 1$ value (level of fixpoint operator nesting; odd number always corresponds to least fixpoints) can be reached via θ by remaining below $2k + 2$.

Theorem (Completeness)

For this constructed ranked predicate abstraction \aleph_θ we have $(M_{\aleph_\theta}, (h_\theta(s), q, g)) \models \phi$ whenever θ is winning for (s, q) .

Conclusion

- Development of extended predicate abstraction that is sound, precise, incremental, and complete for the full mu-calculus (i.e. liveness properties are adequately handled).
- Good foundation for the automated synthesis of abstractions and counter-example-guided abstraction-refinement for branching time.
- Application: extension of existing tools like BLAST or SLAM.