

Consistent Partial Model Checking

Michael Huth

Department of Computing

Imperial College London

Shekhar Pradhan

Department of Computer Science

Vassar College

Thursday, July 10, 2003 at DIMACS

Acknowledgments: Glenn Bruns, Patrice Godefroid, Rahda Jagadeesan, and David Schmidt.

Objectives of talk

1. Sketch model-checking framework that
 - deals with **inconsistencies**
 - allows for **under-specification**
 - has sound notions of **refinement** and abstraction
 - allows for **multiple viewpoints** and
 - **re-uses** existing model checkers whenever possible.
2. Compare this to work on multiple-valued model checking.

Threats to consistency

- distributed contents
- multiple viewpoints
- over-constrained system
- system evolution
- hidden assumptions
- emotionally driven judgments
- etc.

Treating Inconsistency

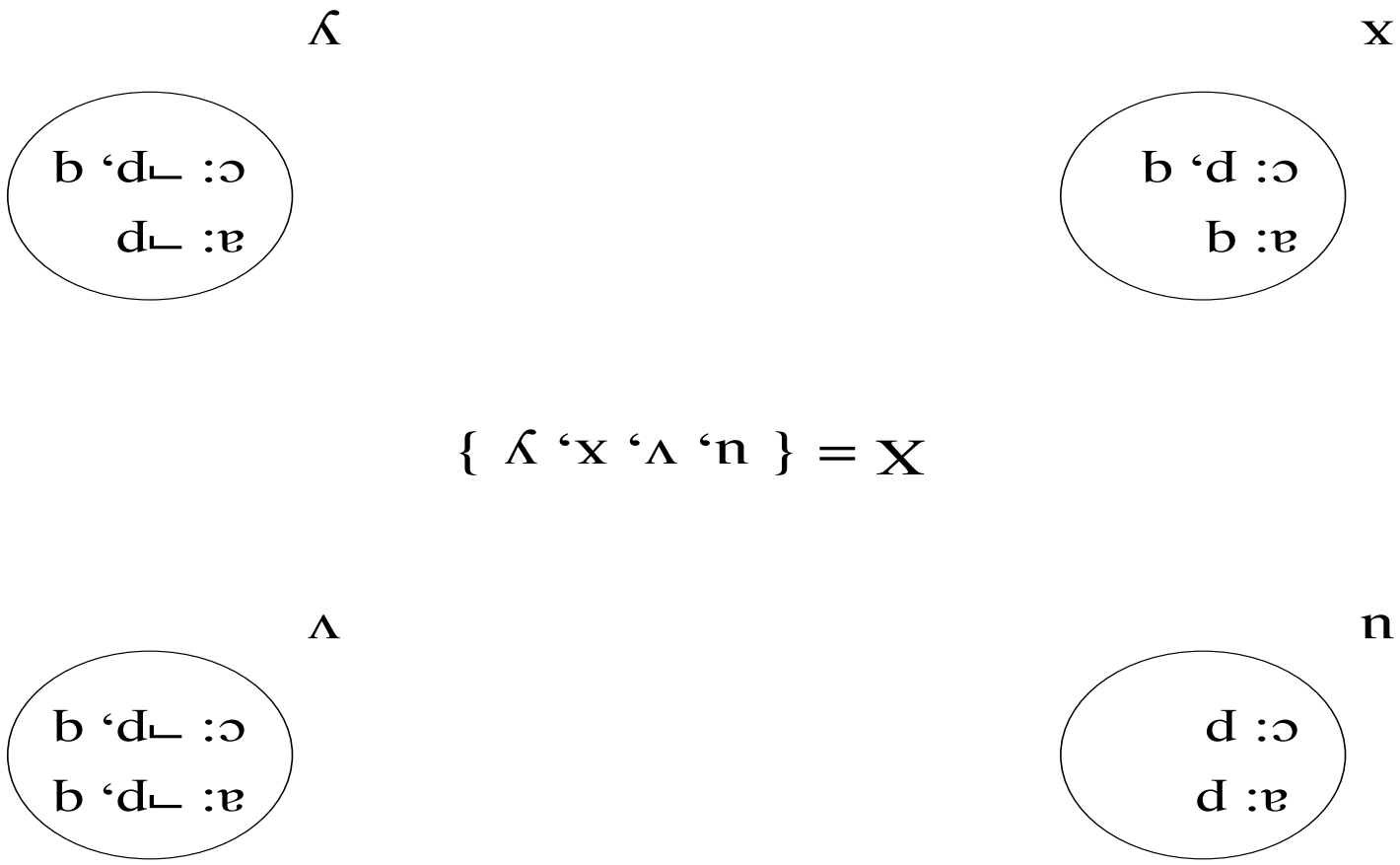
Based on [Nuseibeh, Kramer, and Finkelstein 1994]:

- **detect** inconsistencies
- locate **sources** of inconsistencies
- negotiate **conflict resolution**
- **repair** inconsistencies if possible & feasible
- **mitigate** effect of inconsistencies
- etc.

Bundled models

- Models P comprise finite bundles: $P = (P_x)_{x \in X}$.
- Each P_x model in some ontology.
- Each $x \in X$ may stand for
 - an application aspect
 - a stake-holder's viewpoint
 - a component's version number
 - a model's "information horizon"
 - a domain-specific expert [Fitting 1992]
 - etc.

Legend: rules x : “ x is consistent” and “ b is valid assertion” etc.



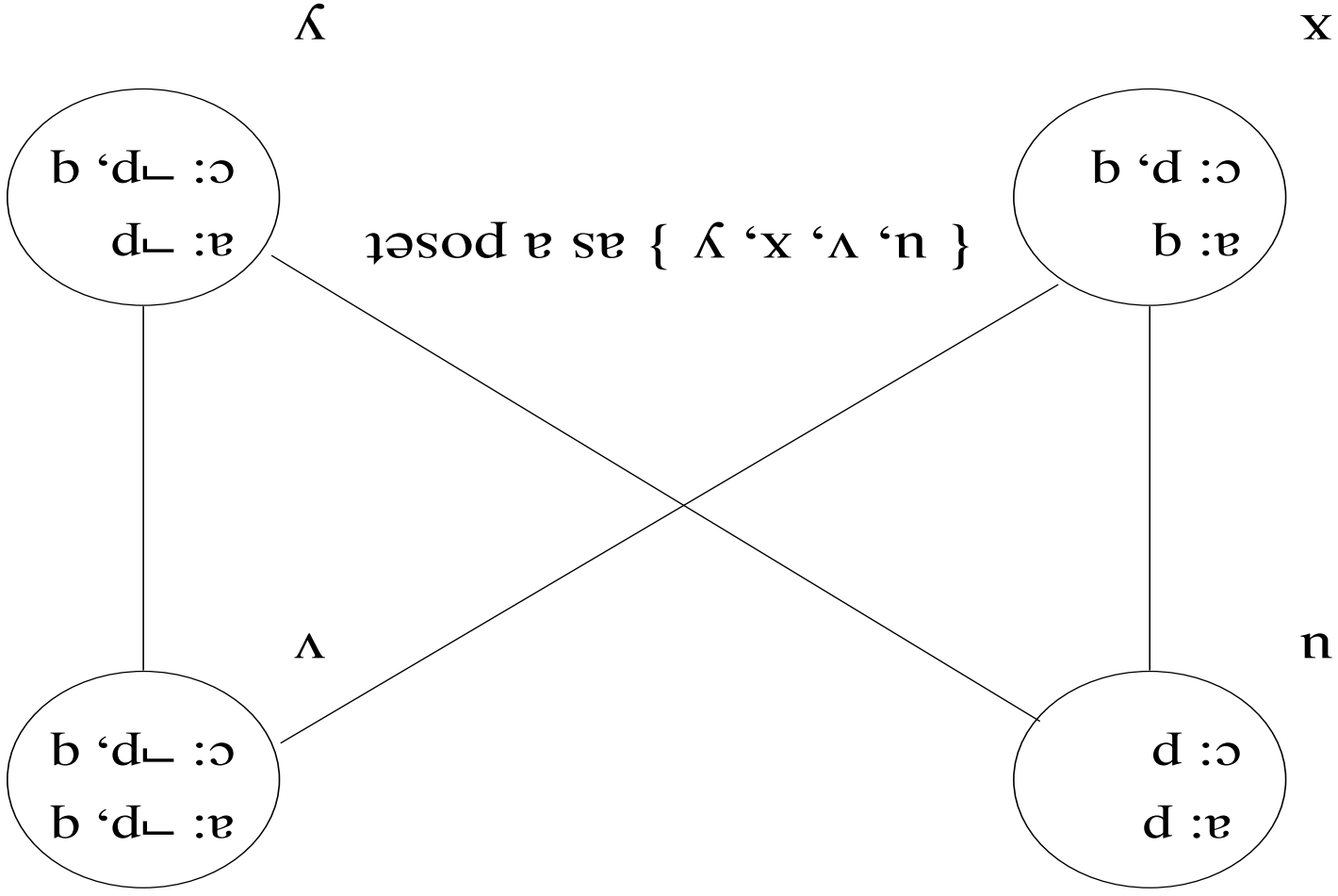
Example of bundled model

Priorities for bundles

1. Models $P = (P^x)_{x \in X}$ bundled over partial order (X, \leq) of priorities [Fitting'92: "dominance"]
2. $x \leq y$ varying interpretations:

- aspect x' more crucial than aspect x
- expert x' more authoritative than expert x
- version x' more recent than version x
- x' contains the information horizon of x
- etc.

Prioritized example



Legend: both u and v have higher priority than x and y .

Partiality of bundles

- Assume range of queries $\phi \in \mathcal{L}$.

- Each P_x answerable to each query $\phi \in \mathcal{L}$.

- Two kinds of answers:

- $P_x \models_a \phi$: x rules “ ϕ valid assertion in P_x ” and
- $P_x \models_c \phi$: x rules “ ϕ consistent in P_x .”

- \hat{Q}_x refines P_x iff P_x abstracts \hat{Q}_x iff

$$(Ass) \quad \forall \phi \in \mathcal{L} : P_x \models_a \phi \Leftrightarrow \hat{Q}_x \models_a \phi.$$

- If “ $P_x \models_a \phi$ and $P_x \not\models_c \phi$ ” for no ϕ and x , the latter means

$$(Con) \quad \forall \phi \in \mathcal{L} : \hat{Q}_x \models_c \phi \Leftrightarrow P_x \models_c \phi.$$

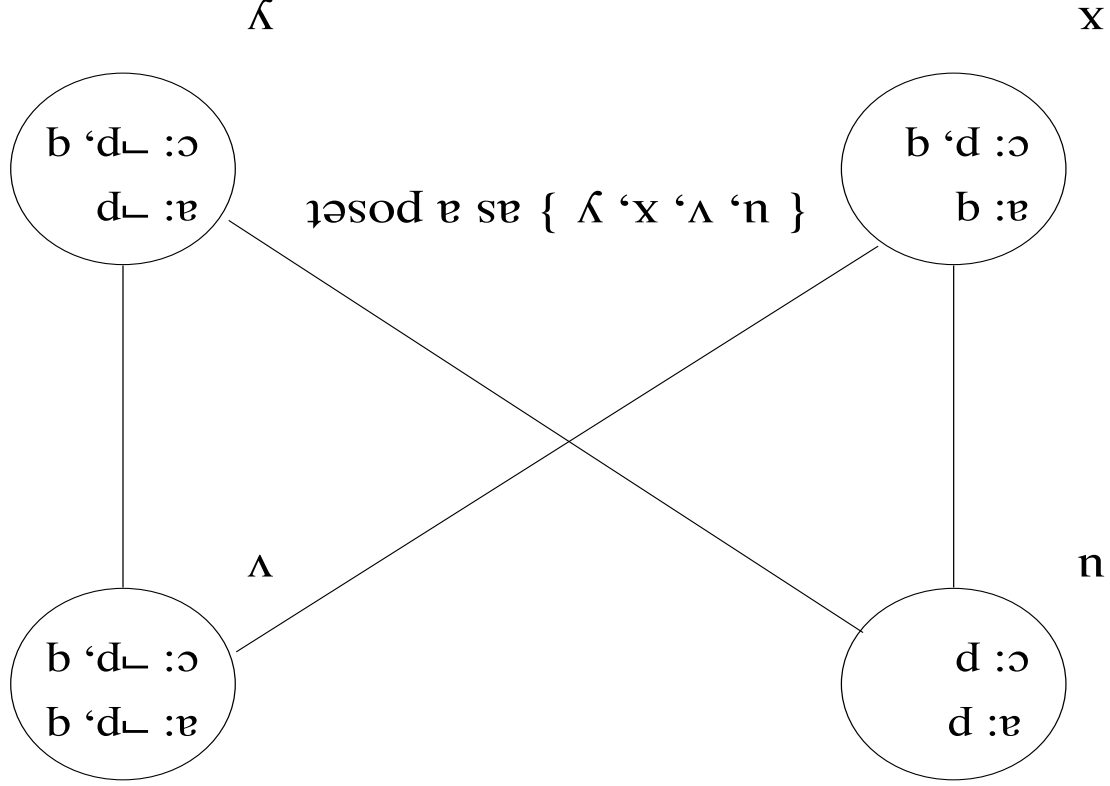
Obligation cones

Platonic bundles:

- defined by $\forall x, y \in X : x \leq y \Leftrightarrow P_x \text{ refines } P_y$
 - have no inconsistencies **across** model boundary \leq
 - ideal **reference frame** against which to detect and locate inconsistencies
- \Leftrightarrow results in “flow laws” of obligations:

$$\begin{aligned} \text{(Ass)} \quad \{X : \phi\}_a \stackrel{\text{def}}{=} \{x \in X \mid \exists y \in X : x \leq y, P_y \models_a \phi\} \\ \text{(Con)} \quad \{X : \phi\}_c \stackrel{\text{def}}{=} \{x \in X \mid \exists y \in X : x \leq y, P_y \models_c \phi\}. \end{aligned}$$

Example cones of influence



$$\{X: d\}_a^a = \{v \in X \mid \exists w \in X : v \leq w, P_w \models_a d\} = \{u, x, y\}$$

$$\{X: \neg d\}_c^c = \{v \in X \mid \exists w \in X : w \leq v, P_w \models_c \neg d\} = \{u, v, y\}.$$

Computing obligation cones

```
if (m == a) { leq = <=; } else { leq = >=; }
U = X; L = emptyset;
while (U != emptyset) {
  for all x in max(U,leq) {
    if (check(P_x,phi,m)) { P_x |= ~m phi ?
      let A = { y | y leq x } in {
        L = L union A;
        U = U \ A;
      }
    } else {
      U = U \ { x };
    }
  }
}
return L;
```

Comments on algorithm

- not compositional in ϕ
- **parametric** in mode m and local checks $\text{check}(P_x, \text{ph}_i, m)$
- complexity \leq complexity for local checks times $|X|$
- parametricity in local checks allows for
 - check being an abstract interface
 - different local **ontologies**
 - **re-use** of model checkers, e.g. [Brunns & Godefroid 2000]
 - **semi-formal** or informal expert rulings.

Detecting and locating inconsistencies

For finite $\mathcal{L} \subseteq \{a, c\}$:

$$\bigcup_{\phi \in \Phi} \{X : \phi\} \not\subseteq \{X : \bigvee \Phi\}$$

$$\bigcup_{\phi \in \Phi} \{X : \phi\} \not\subseteq \{X : \bigwedge \Phi\}$$

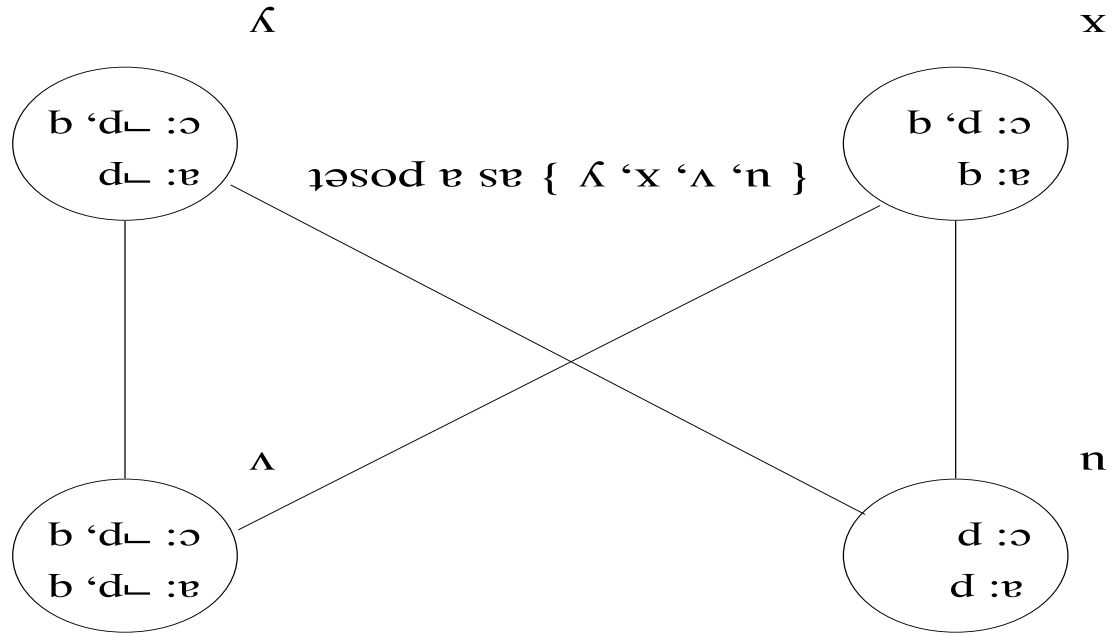
$$\bigcup_{\phi \in \Phi} \{X : \phi\} = \{X : \bigvee \Phi\}$$

Use \bigcup , \bigcap , and \setminus to **instrument** detection and location queries.

E.g.

$\bigcup_{\phi \in \Phi} \{X : \phi\} = \{x \in X \mid x \text{ obliged to hold } \bigvee \Phi \text{ consistent}\}$.

Example instrumentation



1. Detection of inconsistency:

$$\{X : d\}^c \cup \{X : \neg d\}^c = \{n, v, x\} \cup \{n, v, y\} = \{n, v\}.$$

2. Location of inconsistency for n : product of all minimal witnesses for membership; here $\{(x, y)\}$ only.

3. Conflict resolution: experts of all tuples re-assess.

Primer on obligation cones

- all P_x consistent \Leftrightarrow cones are internally consistent:

$$\{X : \phi\}_a \uparrow = (\{X : \phi\}_a \cup \{X : \phi\}_c)$$

- all P_x consistent and P Platonic $\Leftrightarrow \{X : \phi\}_a \subseteq \{X : \phi\}_c$

- $\forall x \in X : Q_x$ refines $P_x \Leftrightarrow$ obligation cones sound under

refinement:

$$\{X_P : \phi\}_a \subseteq \{X_Q : \phi\}_a \quad \& \quad \{X_Q : \phi\}_c \subseteq \{X_P : \phi\}_c$$

- mixed power-domain [Gunter 1992] & [Heckmann 1993] of (X, \leq) right concept for “internal consistency” and “soundness of refinement.”

Multiple-valued model checking

- **compositional semantics** $\llbracket \phi \rrbracket$ element of finite distributive lattice D e.g. [Chechik et al. 2001], i.e.

$\llbracket \phi \rrbracket$ lower set in $(X, \leq) \stackrel{\text{def}}{=} \text{primes of } D$.

- **observables of model** (transitions etc) lower sets of (X, \leq) ;

- **amend model**: **observables pairs** (L, U) of lower and upper

sets of (X, \leq) (resp.); **compositional semantics in two modes**

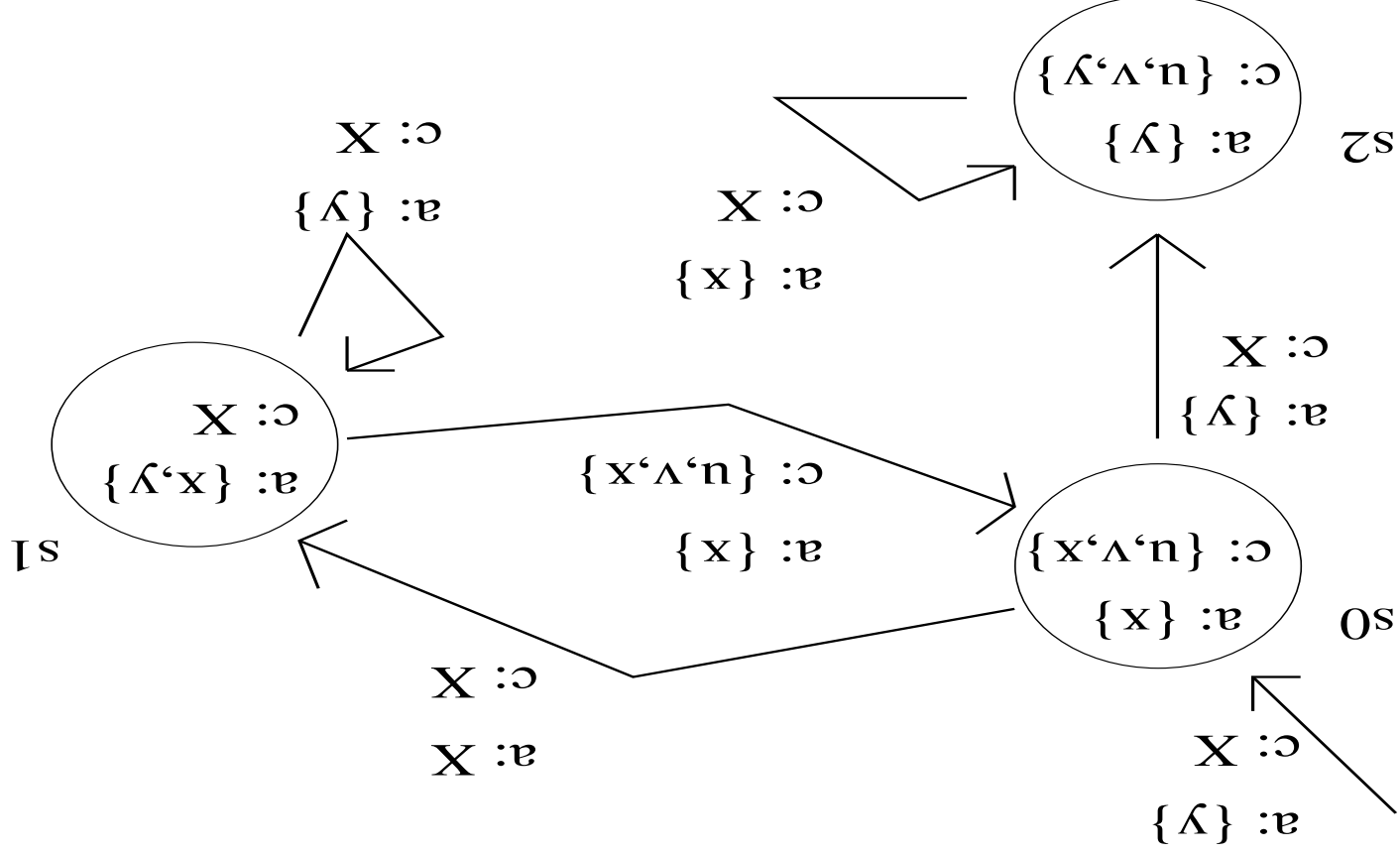
$\llbracket \phi \rrbracket_m, m \in \{c, a\}$;

- **amended model** \rightsquigarrow **projected bundled model** $P = (P^x)_{x \in X}$;

- P **Platonic** \Leftrightarrow

$\{X : \phi\}_m = \llbracket \phi \rrbracket_m$ for all ϕ and m .

Example: multiple-valued modal state machines



Legend: X as before; " s_0 initial state" consistent for all, but valid for only, etc; single predicate p valid for only at s_0 , consistent for all but y at s_0 etc.

Compositional query semantics

$$\begin{aligned} \llbracket \perp \rrbracket_P^d &= (\{\}, \{\}) \\ \llbracket p(u) \rrbracket_P^d &= \text{Lb}(p, d(u)) \\ \llbracket +R(u, v) \rrbracket_P^d &= R_+(d(u), d(v)) \\ \llbracket R(u, v) \rrbracket_P^d &= R(d(u), d(v)) \\ \llbracket \neg \phi \rrbracket_P^d &= (X \setminus \llbracket \phi \rrbracket_{P^a}^d, X \setminus \llbracket \phi \rrbracket_{P^c}^d) \\ \llbracket \phi \vee \psi \rrbracket_P^d &= \llbracket \phi \rrbracket_P^d \sqcup \llbracket \psi \rrbracket_P^d \\ \llbracket \exists u \phi \rrbracket_P^d &= \bigsqcup_{s \in S} \llbracket \phi \rrbracket_{P^{[u \mapsto s]}}^d. \end{aligned}$$

- $s \in S$ set of states and $\llbracket \phi \rrbracket_P^d = (\llbracket \phi \rrbracket_{P^a}^d, \llbracket \phi \rrbracket_{P^c}^d)$;

- $p \in AP$ state predicates; $u, v, \dots \in \text{Var}$ variables;

- $R(s, s')$ (one transition), $R_+(s, s')$ (one or more transitions),

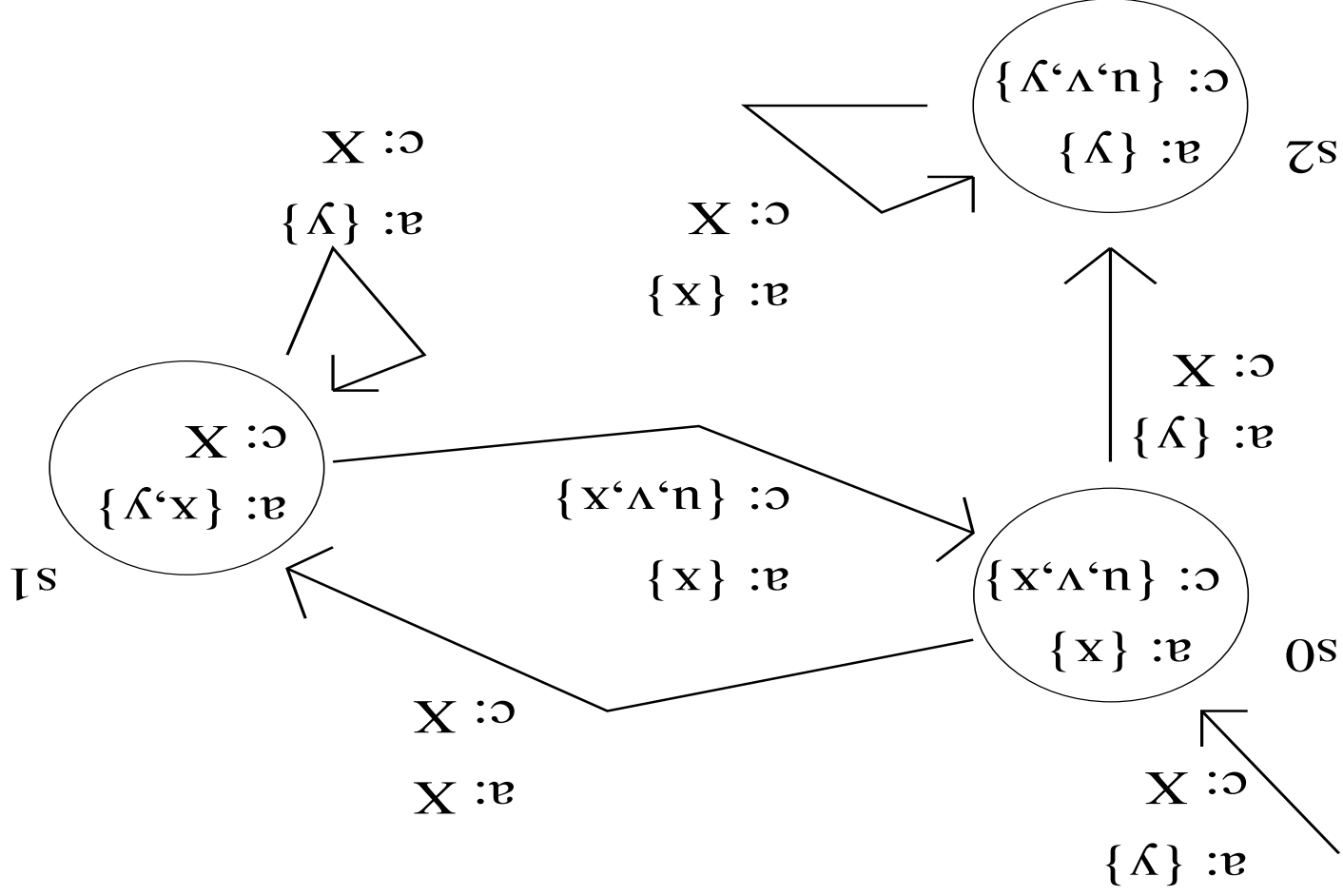
and $\text{Lb}(p, s)$ (labeling) all of the form (T, U) with T lower set, U upper set, and $T \bar{\subseteq} U$;

- $\rho: \text{Var} \rightarrow S$ environment; and

- $\bigsqcup_{i \in I} (T_i, U_i) \stackrel{\text{def}}{=} (\bigcup_{i \in I} T_i, \bigcup_{i \in I} U_i)$.

A model check

Consider $\phi \stackrel{\text{def}}{=} \exists v (+R(v, v) \wedge \neg p(v))$: “There is a state that is on a cycle and does not satisfy p .”



Determining bundled model

- The formula

$$\phi \stackrel{\text{def}}{=} \exists v (+R(v, v) \vee \neg p(v))$$

- is closed (p not a variable) \Leftrightarrow checks independent of p .
- In P_x every observable (L, U) turns into observable

1;	if $x \in L$ (valid in P_x)
0;	if $x \notin U$ (inconsistent in P_x) and
1/2;	otherwise (consistent but invalid in P_x) .

\Rightarrow each P_x three-valued model.

Technical reference

M. Huth and S. Pradhan, *Consistent partial model checking*,
Electronic Notes in Theoretical Computer Science 73 (2003),
URL: <http://www.elsevier.nl/locate/entcs/volume73.html>,
39 pages.

Conclusions

(This page is intentionally left blank.)