

---

# Model Checking with Multi-Valued Logics

Glenn Bruns

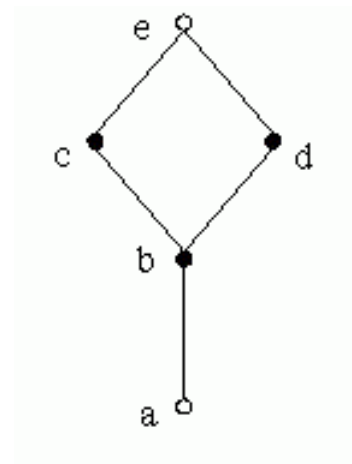
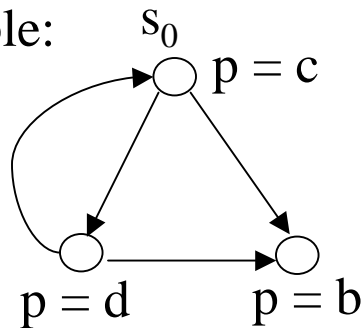
Patrice Godefroid

Bell Laboratories, Lucent Technologies

# Multi-Valued Model Checking: Definition

- Kripke structure  $K$  where, in every state, every atomic proposition is mapped to an element of a lattice  $L$ .

– Example:



- Multi-valued temporal logic:

– Syntax: unchanged

– Semantics: unchanged except  $\wedge$  is meet in  $L$ ,  $\vee$  is join in  $L$

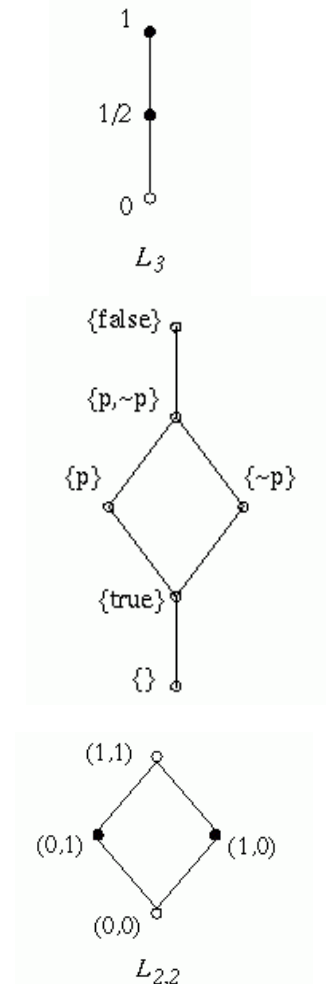
– Example:  $\phi = p \wedge EX p$

- $[K \models \phi]$  (“model checking”) returns a value in  $L$ .

– Example:  $[(K, s_0) \models p \wedge EX p] = (c \wedge (d \vee b)) = b$

# Motivation: Applications

- Model Checking using 3-valued abstractions
  - Automatically abstract a program into a 3-valued model  $K$
  - Check any temporal property  $\phi$  on this model
  - If  $[K \models \phi] = \frac{1}{2}$ , refine the model and repeat the process
- Temporal-logic query checking
  - Given a “query”  $\phi$  (ex:  $AG?$ ), what is the set of strongest propositional formulas  $f$  (built from  $P$ ) such that  $K \models \phi[? \leftarrow f]$
- Multi-viewpoints model checking
  - What properties do different experts agree on?
- All these problems reduce to “multi-valued model checking”



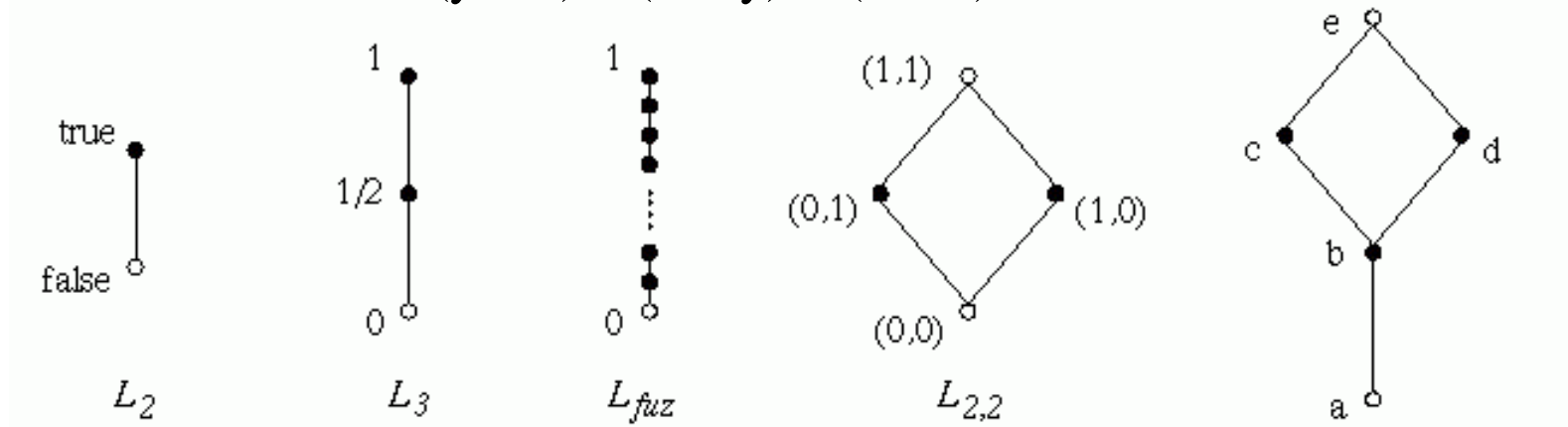
# Two Approaches

---

- By reduction
  - Idea: reduction to several standard, 2-valued model-checking problems
  - Advantage: re-use of existing model checkers
  - New result: simple and general method for reduction
- Direct (automata-theoretic) approach
  - Idea: represent the formula by an EAA, and compute product with  $K$
  - Advantage: works in a more “demand-driven” way
  - New result: maximum-value theorem for EAA and general automata-theoretic approach to multi-valued model checking

# Lattices and Negation

- We consider finite (hence complete) distributive lattices.
  - Complete:  $\forall X \subseteq L: \bigwedge\{X\}$  and  $\bigvee\{X\}$  exist in  $L$
  - Distributive:  $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$



- A *join-irreducible element*  $x$  of a distributive lattice  $L$  is an element that is not  $\perp$  and for which  $(x = y \vee z) \Rightarrow (x = y \text{ or } x = z)$
- DeMorgan lattice: every  $x \in L$  has a unique complement  $\neg x$  such that  $\neg\neg x = x$ , DeMorgan's laws hold, and  $(x \leq y) \Rightarrow (\neg y \leq \neg x)$

# Reduction Method (Approach 1)

---

- Given  $K, \phi \in \mu$ -calculus, and a finite distributive DeMorgan  $L$ :
  - Push  $\neg$  inwards (using DeMorgan laws) to get  $\phi$  in positive normal form.
  - $\forall x \in L$ , define  $K_x$  as  $K$  except that  $\theta_x(s)(p) = \theta(s)(p) \geq x$
  - Let  $J(L)$  denote the (finite) set of join-irreducible elements of  $L$ .
  - Lemma 1: Given  $K$  over  $L$ ,  $s$  in  $K$ ,  $x \in J(L)$ :  $(K_x, s) \models \phi \Leftrightarrow x \leq [(K, s) \models \phi]$
  - Theorem 1:  $[(K, s) \models \phi] = \bigvee \{x \in J(L) \mid (K_x, s) \models \phi\}$
  - Theorem 2: Given a TL, multi-valued model checking  $[(K, s) \models \phi]$  for TL has the same complexity in  $K$  and  $\phi$  as traditional model checking for TL, and can be done in time  $O(|J(L)|)$ .
- Notes:
  - Sometimes complexity in  $|J(L)|$  is better than linear...
  - These results can easily be extended to multi-valued transitions...

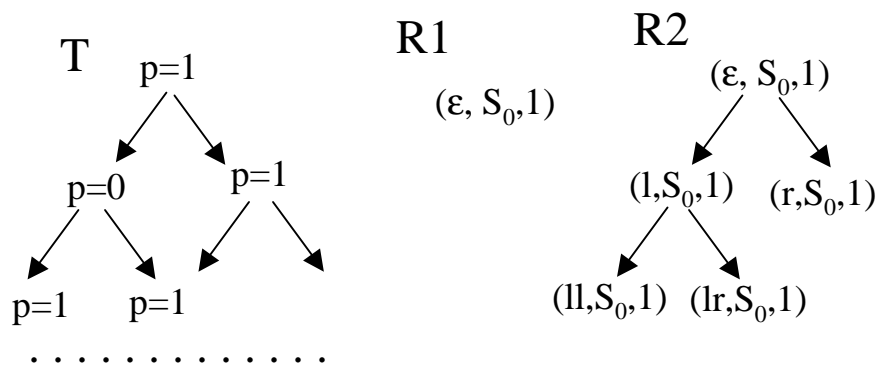
# Comparison with Related Work

---

- Generalizes reduction methods for specific lattices
  - 3-valued model checking [BrunsGodefroid00]
  - Several other lattices [KonikowskaPenczek02]
- Simplifies other reduction method using join-irreducible elements
  - [GurfinkelChechik03]
- Extends work on “many-valued modal logics” [Fitting92]
  - Reduction to standard Kripke structure vs. “multi-expert models”
  - Join-irreducible elements instead of “proper-prime filters”
  - Fixpoint modal logic vs. modal logic
  - Different treatment of negation (DeMorgan lattices vs. Heyting algebras)
- Different from work on “AC-lattices” [HuthPradhan02]

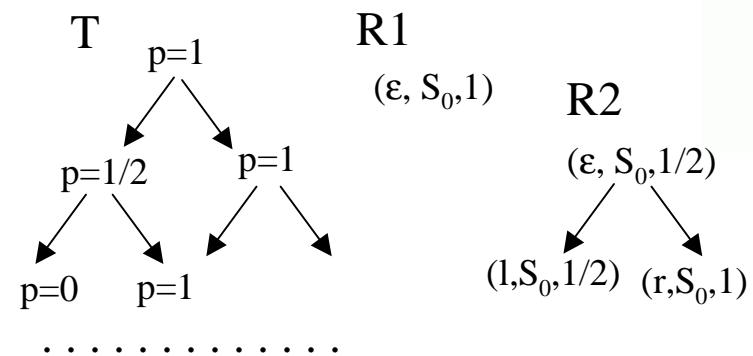
# Extended Alternating Automata (Approach 2)

- Alternating Automaton  $A=(\Sigma,S,s_0,\rho,F)$  with input alphabet  $\Sigma$ , transition function  $\rho$ , acceptance condition  $F$
- Ex:  $\rho(s_0,\sigma,2)=\sigma(p) \vee ((l,s_0)\wedge(r,s_0))$  and  $F=\{\}$  (equivalent to  $AFp$  in CTL)
- Run:  $\infty$  input tree  $T \rightarrow$  run tree  $R$



- $T$  is accepted by  $A$  (denoted  $T \in L(A)$ ) if  $A$  has an accepting run  $R$  on  $T$ :
  - every  $\infty$  branch of  $R$  satisfies  $F$ .

- Extended Alternating Automaton [BrunsGodefroid01]: same as AA except  $\rho$  is defined on  $L$  with  $\wedge$  and  $\vee$
- Run:  $\infty$  input tree  $T \rightarrow$  run tree  $R$  labeled with non- $\perp$  elements of  $L$



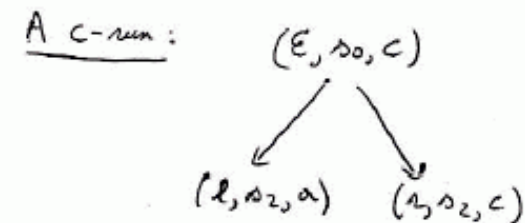
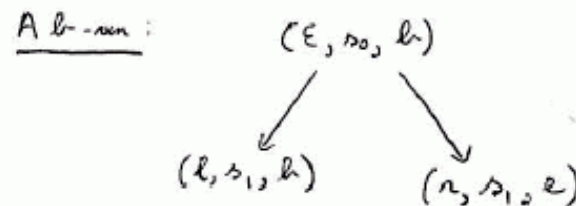
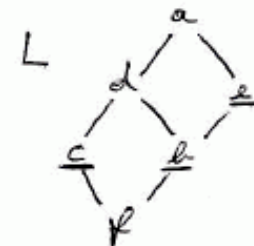
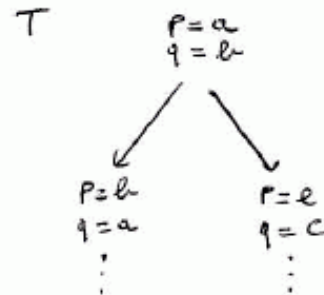
- $T$  is accepted by  $A$  with value  $v$  (denoted  $T \in L_v(A)$ ) if  $A$  has an accepting  $v$ -run  $R$  on  $T$ :
  - $v$  labels the root node of  $R$
  - every  $\infty$  branch of  $R$  satisfies  $F$ .



# Maximum-value Theorem

- Thm: Let  $A$  be a finite tree EAA over  $L$ , and let  $T$  be an infinite input tree. Then the subset  $\{v \mid T \in L_v(A)\}$  of  $L$  has a maximum value  $\text{Max}(A, T)$ .
- Note: nontrivial!

$$\begin{aligned} \text{EAA: } P(s_0, \sigma, z) &= ((l, s_1) \vee (r, s_2)) \wedge ((l, s_1) \vee (r, s_2)) \\ f(s_1, \sigma, z) &= \sigma(P) \\ p(s_2, \sigma, z) &= \sigma(Q) \end{aligned}$$



**BUT** THERE IS NO RUN WITH VALUE  $bvc = d$  (!)

# Proof Idea

---

- Define a lattice  $V$  of valuation trees ordered by the sub-tree relation and  $\leq$  on  $L$ . Since  $L$  is complete,  $V$  is a complete lattice.
- Define a function  $F: V \rightarrow V$  that computes the transition function  $\rho$
- Runs correspond to fixpoints of  $F$ .
- Apply Knaster-Tarski's theorem to  $F$  (order-preserving on  $V$ ):  
“the join  $R$  of all runs (fixpoints of  $F$ ) is a run (fixpoint of  $F$ ).”
- Problem:  $R$  may not be accepting! (since the join of infinitely-many finite paths may not be accepting...)
- Solution: provide a construction to eliminate all infinite non-accepting paths in  $R$  while preserving the label of its root node...

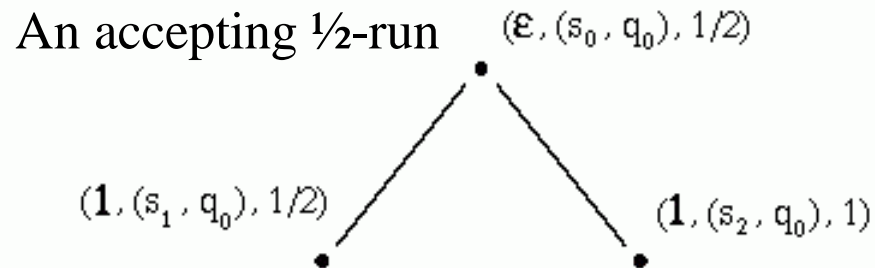
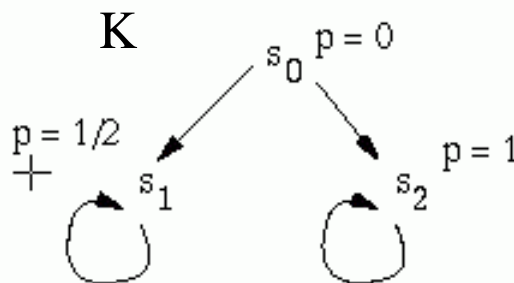
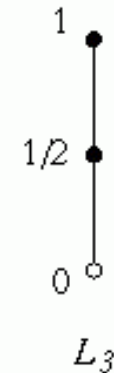
# Model Checking with EAA

---

- Automata-theoretic approach to multi-valued model checking (extends [KupfermanVardiWolper00]):
  - Translate  $\phi$  into a tree EAA  $A_\phi$  such that  $[T \models \phi] = \text{Max}(A_\phi, T)$   
(translation similar to the traditional one except for atomic propositions)
  - Compute the product  $A_{K,\phi}$  of  $K$  and  $A_\phi$  (a word EAA on 1-letter alphabet)
  - Theorem:  $[K \models \phi] = \text{Max}(A_{K,\phi})$
  - Computing  $\text{Max}(A_{K,\phi})$  has the same complexity in  $|A_{K,\phi}|$  as checking language emptiness in regular word AA on 1-letter alphabet, and can be done in time  $O(|h(L)|)$ .
  - Example: if Buchi acceptance condition, quadratic time in  $|A_{K,\phi}|$ , or even linear time in  $|A_{K,\phi}|$  if the EAA is also ‘weak’ (e.g., for CTL).

# Example

- Consider the lattice  $L_3$
- Consider the formula  $\phi = \text{AF } p$  ( $= \mu X.p \vee \square X$ )
- $A_\phi$  is a tree EAA on  $L_3$  with  $\rho(q_0, \sigma, 2) = \sigma(p) \vee ((l, q_0) \wedge (r, q_0))$  and  $F = \{ \}$
- Given  $K$  below,  $A_{K, \phi}$  is a word EAA on 1-letter alphabet  $\{a\}$  with  $\rho((s_0, q_0), a, 1) = 0 \vee ((s_1, q_0) \wedge (s_2, q_0))$ ,  $\rho((s_1, q_0), a, 1) = 1/2 \vee (s_1, q_0)$ ,  $\rho((s_2, q_0), a, 1) = 1 \vee (s_2, q_0)$ , and  $F = \{ \}$
- $[K \models \phi] = \text{Max}(A_{K, \phi}) = 1/2$



# Summary and Conclusions

---

- Summary: two approaches to multi-valued model checking
  - By reduction
    - Advantage: re-use of existing model-checking tools
    - New result: simple and general method based on join-irreducible elements for finite distributive DeMorgan lattices and full  $\mu$ -calculus
  - Direct, automata-theoretic
    - Advantage: more “on-the-fly”/demand-driven
    - New result: maximum-value theorem for EAA and general automata-theoretic approach for DeMorgan lattices and full  $\mu$ -calculus
- Future work:
  - Complementation of EAA...
  - Detailed study of algorithms for computing  $\text{Max}(\text{EAA})$  (infinite games + lattice equations)...
  - Other applications: quantitative games for resource optimization?