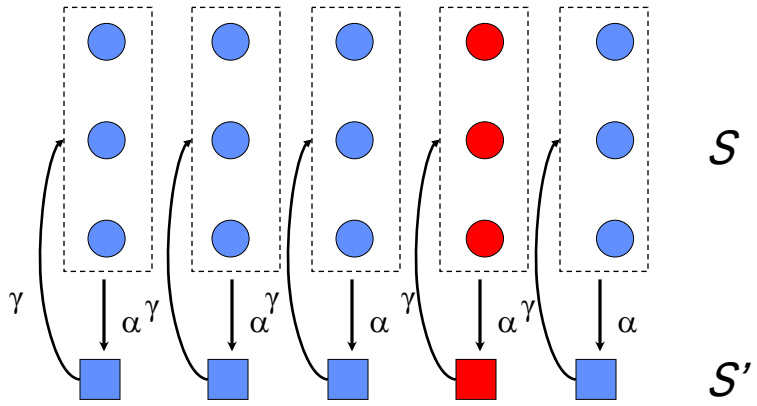
Abstraction (Cont'd)

- ✓ Defining an Abstract Domain
  - ✓ variable elimination, data abstraction, predicate abstraction
- ✓ Abstraction for Universal/Existential Properties
  - ✓ over- and under-approximations
- ➲ Abstraction for Mixed Properties
  - ↳ 3-valued abstraction
- ➲ Overlapping Abstract Domains
  - ↳ Belnap (4-valued) abstraction

1

Recall: Defining an Abstract Domain



Abstraction $\alpha : S \rightarrow S'$ Concretization $\gamma : S' \rightarrow 2^S$

2

Abstract Kripke Structure

- Abstract interpretation of atomic propositions
  - $I'(a, p)$ = true iff forall s in $\gamma(a)$, $I(s, p)$ = true
  - $I'(a, p)$ = false iff forall s in $\gamma(a)$, $I(s, p)$ = false

- Abstract Transition Relation (2 choices)
  - Over-Approximation (Existential)
    - Make a transition from an abstract state if *at least one* corresponding concrete state has the transition.
  - Under-Approximation (Universal)
    - Make a transition from an abstract state if *all* the corresponding concrete states have the transition.

3

Which abstraction to use?

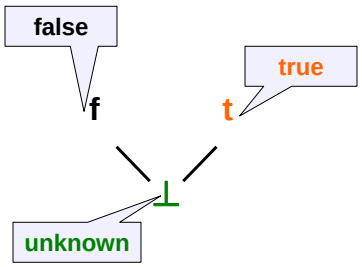
Property Type	Expected Result	Abstraction to use
Universal (ACTL, LTL)	True	Over-
	False	Under-
Existential (ECTL)	True	Under-
	False	Over-

But what about mixed properties?!

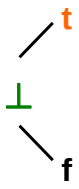
4

3-Valued Kleene Logic

Information Ordering



Truth Ordering



$t \wedge \bot = \bot$

$t \vee \bot = t$

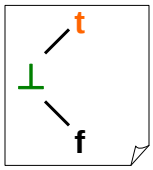
$\neg t = f$

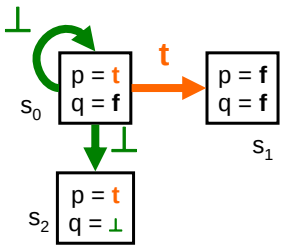
$\neg \bot = \bot$

5

3-Valued Kripke Structures

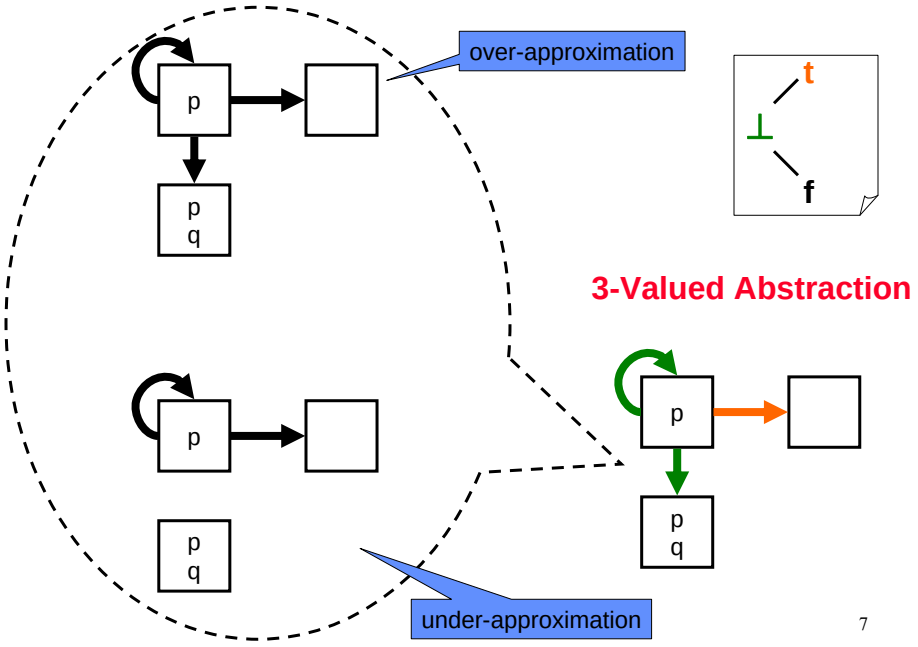
- Kripke structures extended to 3-valued logic
- Propositions can be
  - True, False, or Unknown
- Transitions
  - possible: ⊥
  - necessary and possible: t
  - impossible: f





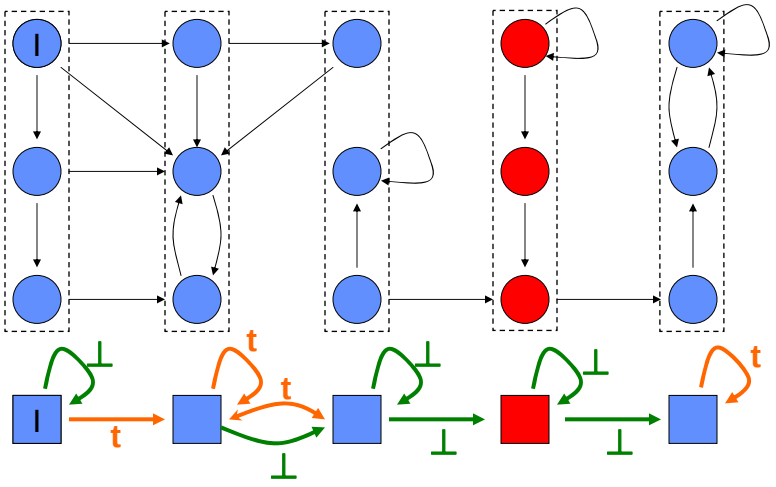
6

3-Valued Abstraction



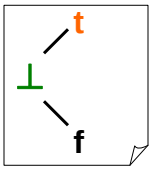
7

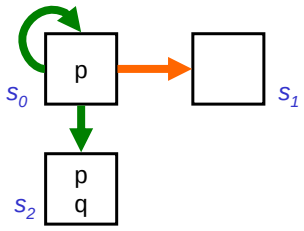
Example Revisited (3-Val Abstraction)



8

Model-Checking with 3 Values





- Usual semantics of temporal operators
- BUT connectives ∧ ∨ ¬ are interpreted in 3-Valued Logic

$$(\mathrm{EX}\ p)(s) = \bigvee_t R(s,t) \wedge p(t)$$

Examples

$(\mathrm{EX}\ \neg p)(s_0) = \mathbf{t}$

$(\mathrm{EX}\ q)(s_0) = \bot$

$(\mathrm{EX}\ \neg p \wedge q)(s_0) = \mathbf{f}$

9

Preservation via 3-Valued Abstraction

- Let φ be a temporal formula (CTL)
- Let K' be a 3-valued abstraction of K
- Preservation Theorem

Abstract MC Result	Concrete Information
True	$K \models \varphi$
False	$K \models \neg\varphi$
Maybe	$K \models \varphi$ or $K \models \neg\varphi$

no information

Preserves truth and falsity of arbitrary properties!

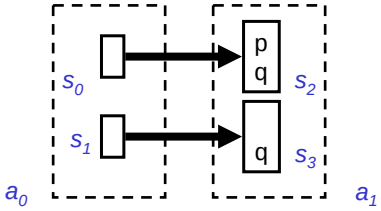
10

Abstraction (Outline)

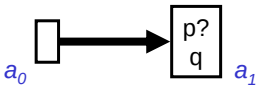
- ✓Defining an Abstract Domain
  - ✓variable elimination, data abstraction, predicate abstraction
- ✓Abstraction for Universal/Existential Properties
  - ✓over- and under-approximations
- ✓Abstraction for Mixed Properties
  - ✓3-valued abstraction
- ➲ Overlapping Abstract Domains
  - ↳Belnap (4-valued) abstraction

11

Example: Coarse Abstract Domain

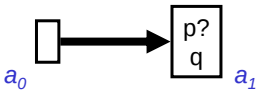


Over-Approximation



AX (p v ¬p) is inconclusive

Under-Approximation

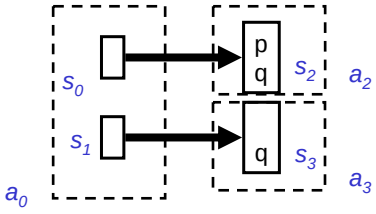


EX (q) is true

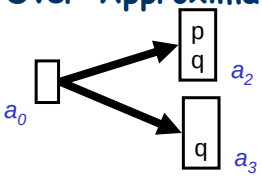
Goal: make AX conclusive as well, via domain refinement

12

Example: Refined Abstract Domain

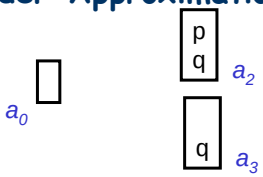


Over-Approximation



AX (p ∨ ¬p) is true

Under-Approximation

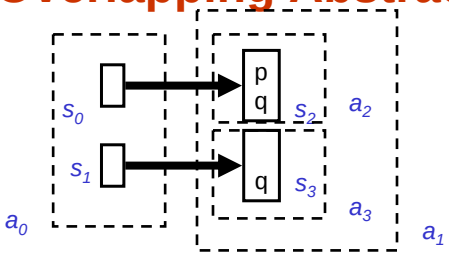


EX (q) is inconclusive

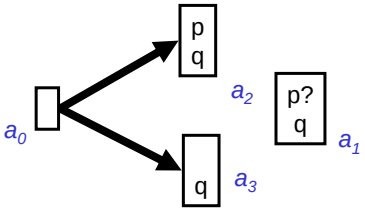
Partitioned domain does not work!
Need an overlapping abstract domain!!!

13

Example: Overlapping Abstract Domain

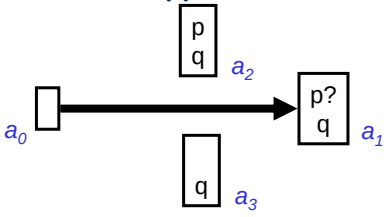


Over-Approximation



AX (p ∨ ¬p) is true

Under-Approximation



EX (q) is true

14

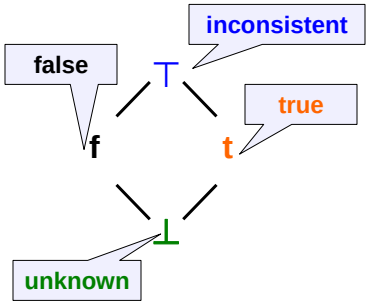
Supporting Overlapping Abstract Domains

- Goal
  - as before, want to combine over- and under-approximations to support analysis of mixed properties
- Problem
  - 3-valued logic is no longer sufficient
  - need to deal with 4 types of transitions
    - over-, under-, both over- and under-, and neither
    - i.e., under-approx is no longer a subset of over-approx
- Solution
  - use 4-valued Belnap logic

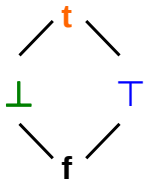
15

Belnap Logic

Information Ordering



Truth Ordering



$t \wedge \bot = \bot$

$t \vee \bot = t$

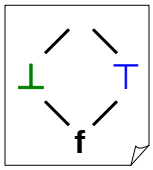
$\neg t = f$

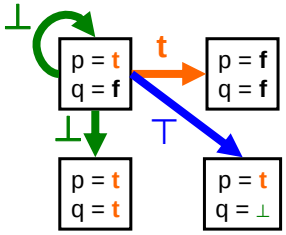
$\neg \bot = \bot$

16

Belnap Kripke Structures

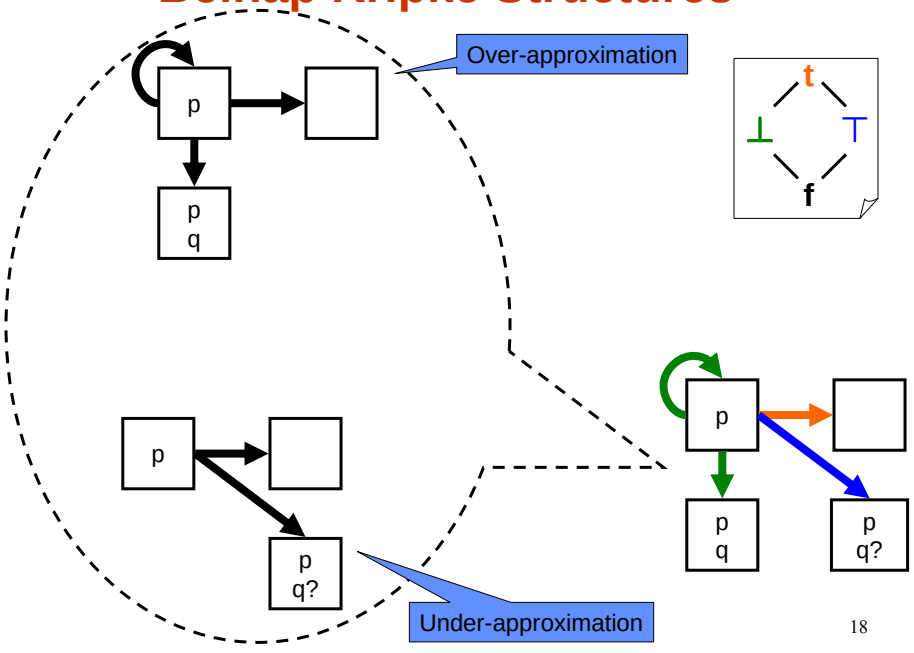
- Kripke structures extended to Belnap logic
- Propositions
  - True, False, or Unknown
- Transitions
  - only under-approximation: ⊤
  - only over-approximation: ⊥
  - both over- and under-: t
  - neither: f





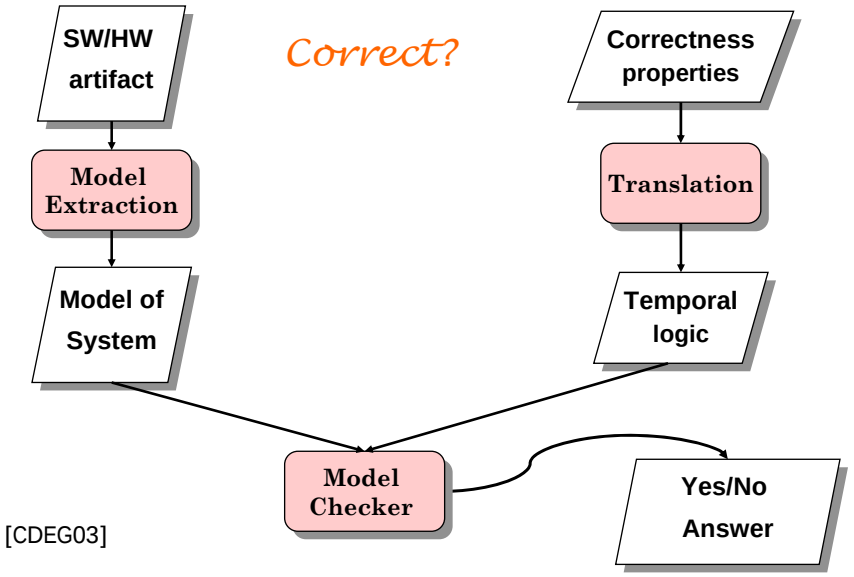
17

Belnap Kripke Structures



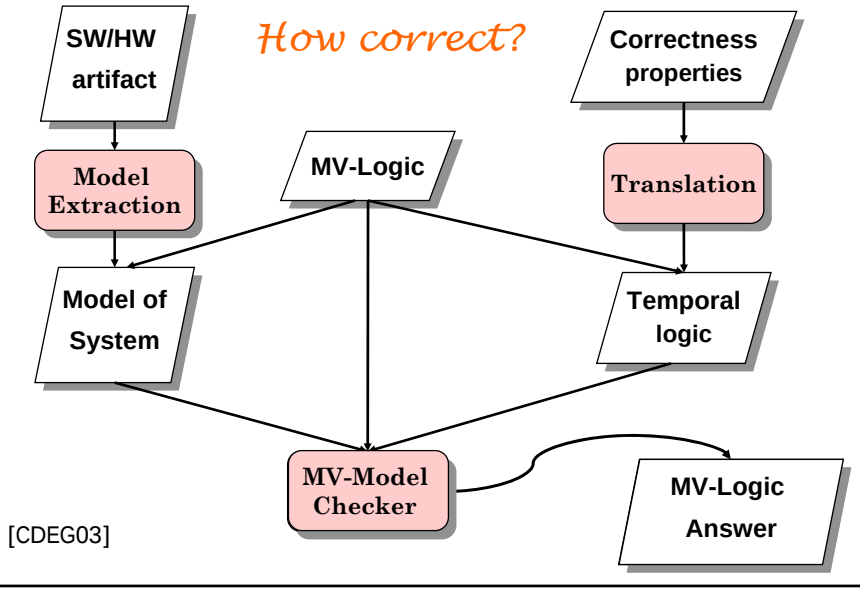
18

Overview of Model Checking



[CDEG03]

Overview of MV-Model Checking



[CDEG03]

Preservation via Belnap Abstraction

- Let φ be a temporal formula (CTL)
- Let *K'* be a Belnap abstraction of *K*
- Preservation Theorem

Abstract MC Result	Concrete Information
True	$K \models \varphi$
False	$K \models \neg\varphi$
$\bot$	$K \models \varphi$ or $K \models \neg\varphi$
$\top$	$K \models \varphi$ and $K \models \neg\varphi$

Not possible for a sound abstraction

Preserves truth and falsity of arbitrary properties!

21

Summary

Abstraction is the key to scaling up

1. **Choose an abstract domain**
   - Variable elimination, data abstraction, predicate abstraction, …
2. **Choose a type of abstraction**
   - Over-, Under-, 3Val, Belnap
3. **Build an abstract model ($$$$$)**
4. **Model-check the property on the abstract model**
5. **If the result is conclusive, STOP**
6. **Otherwise, pick a new abstract domain, REPEAT**

22

References

- [DGG97] D. Dams, R. Gerth, and O. Grumberg, “Abstract Interpretation of Reactive Systems”. In TOPLAS, No. 19, Vol. 2, pp. 253-291, 1997.
- [CDEG03] M. Chechik, B. Devereux, S. Easterbrook, and A. Gurfinkel, “Multi-Valued Symbolic Model-Checking”. In TOSEM, No. 4, Vol. 12, pp. 1-38, 2003.

Acknowledgements

These slides are based on the tutorial “Model-Checking: From Software to Hardware” given by Marsha Chechik and Arie Gurfinkel at Formal Methods 2006.