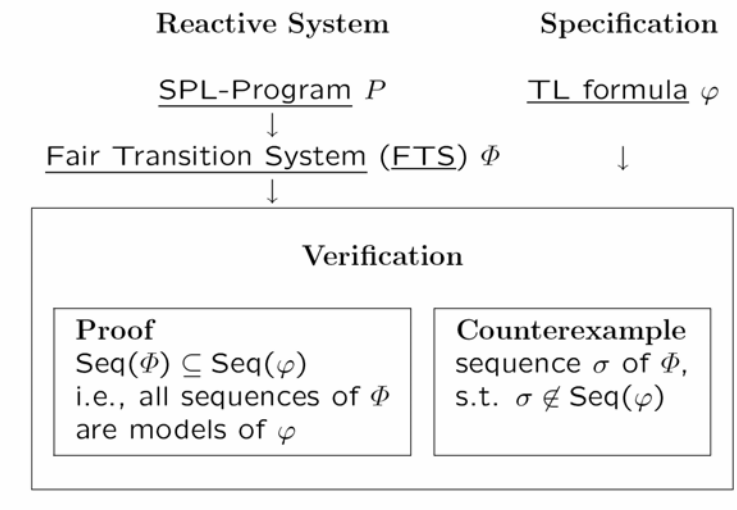
Verification – Lecture 14
Midterm Review

Bernd Finkbeiner – Sven Schewe
Rayna Dimitrova – Lars Kuhtz – Anne Proetzsch

Wintersemester 2007/2008

Verification



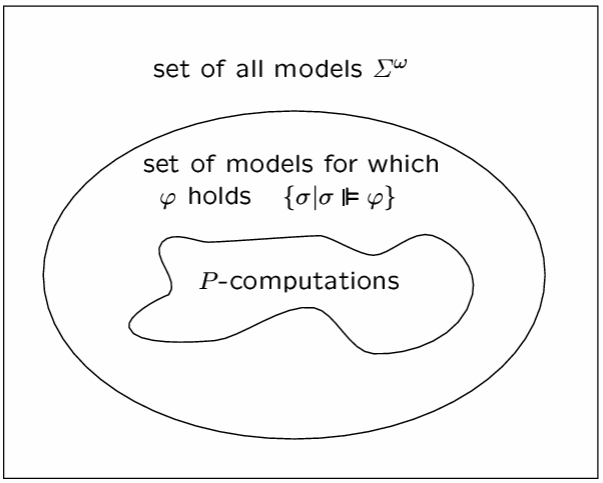
Fair Transition Systems

$$\Phi = (V, \theta, \mathcal{T}, \mathcal{J}, \mathcal{C})$$

- $\mathcal{J} \subseteq \mathcal{T}$: set of just (weakly fair) transitions
- $\mathcal{C} \subseteq \mathcal{T}$: set of compassionate (strongly fair) transitions

- Justice: for each just transition it is not the case that the transition is continually enabled but only taken at finitely many positions.
- Compassion: for each compassionate transition it is not the case that the transition is enabled at infinitely many positions but only taken at finitely many positions.

P-Validity



Validity

	general	program P
state formula q	$\models q$ state valid „q holds in all states"	$P \models q$ P-state valid „q holds in all P-accessible states"
temporal formula φ.	$\models \varphi$ Valid „φ holds in the first position of every sequence"	$P \models \varphi$ P-valid „φ holds in the first position of every P-computation"

Deductive Verification — Model Checking

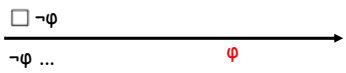
Combination

+ Infinite-state	+	Finite-state −
− Only proofs	+	Counterexamples +
− Interactive	+/−	Automatic +

(undecidable in general)

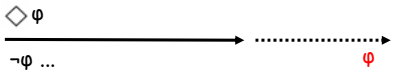
Safety versus Liveness

- "Nothing bad ever happens"
- All finite prefixes satisfy a certain requirement (does not depend on limit behavior)
- Counter-examples are finite



- Provable by induction over reachable states.
- Cannot distinguish runs and computations
- Example:
  □(x=0 → ◯x=0)

- "Something good eventually happens"
- Infinite counter-examples



- Proof requires assumptions about nondeterministic choices (Justice and/or Compassion)
- E□◇le:
  (x=0)

Deductive Verification

- Invariance properties $P \models \Box q$ Safety
- Precedence properties
  $p \Rightarrow q_m \mathcal{W} q_{m-1} \cdots q_1 \mathcal{W} q_0$

Liveness

- Response properties $p \Rightarrow \Diamond q$

Invariance Proofs

- Option 1:
  Proof rules,
  e.g. INV:

For assertions q, φ (complete!)

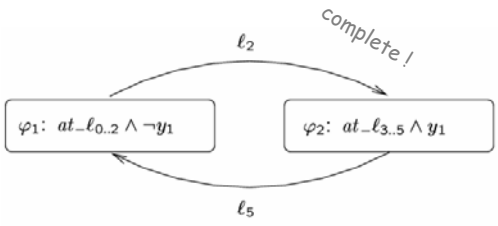
I1. $P \models \varphi \rightarrow q$

I2. $P \models \Theta \rightarrow \varphi$

I3. $P \models \{\varphi\}\ \mathcal{T}\ \{\varphi\}$

$P \models \Box q$ INV

- Option 2:
  Verification diagrams:



Precedence Proofs

- Option 1:
  Proof rules,
  e.g. NWAIT:

Rule NWAIT (nested waiting-for)

For assertions $p,\ q_0, q_1, \ldots, q_m$ and $\varphi_0, \varphi_1, \ldots, \varphi_m$

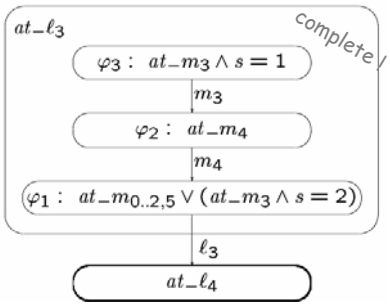
N1. $p \rightarrow \bigvee_{j=0}^{m} \varphi_j$ (complete!)

N2. $\varphi_i \rightarrow q_i$ for $i = 0, 1, \ldots, m$

N3. $\{\varphi_i\}\mathcal{T}\left\{\bigvee_{j \le i} \varphi_j\right\}$ for $i = 0, 1, \ldots, m$

$p \Rightarrow q_m\ \mathcal{W}\ q_{m-1} \cdots q_1\ \mathcal{W}\ q_0$

- Option 2:
  Verification diagrams:



Response Proofs

- Option 1:
  Proof rules,
  e.g. RESP-J:

For assertions p, q, φ, and transition $\tau_h \in \mathcal{J}$,

J1. $p \rightarrow q \vee \varphi$

J2. $\{\varphi\}\ \mathcal{T}\ \{q \vee \varphi\}$

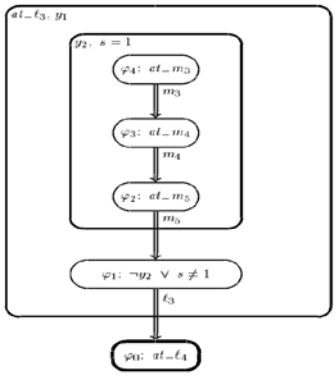
J3. $\{\varphi\}\ \tau_h\ \{q\}$

J4. $\varphi \rightarrow En(\tau_h)$

$$p \Rightarrow \Diamond q$$

NOT complete !

- Option 2:
  Verification diagrams:



Rule WELL-J

For assertions p and $q = \varphi_0, \varphi_1, \ldots, \varphi_m$;
transitions $\tau_1, \ldots, \tau_m \in \mathcal{J}$,
a well-founded domain $(\mathcal{A}, \succ)$, and
ranking functions $\delta_0, \ldots, \delta_m$: $\Sigma \mapsto \mathcal{A}$

JW1. $p \rightarrow \bigvee_{j=0}^{m} \varphi_j$

JW2. $\rho_\tau \wedge \varphi_i \rightarrow \left[\bigvee_{j=0}^{m} (\varphi'_j \wedge \delta_i \succ \delta'_j)\right] \vee (\varphi'_i \wedge \delta_i = \delta'_i)$ for every $\tau \in \mathcal{T}$

JW3. $\rho_{\tau_i} \wedge \varphi_i \rightarrow \bigvee_{j=0}^{m} (\varphi'_j \wedge \delta_i \succ \delta'_j)$

JW4. $\varphi_i \rightarrow En(\tau_i)$

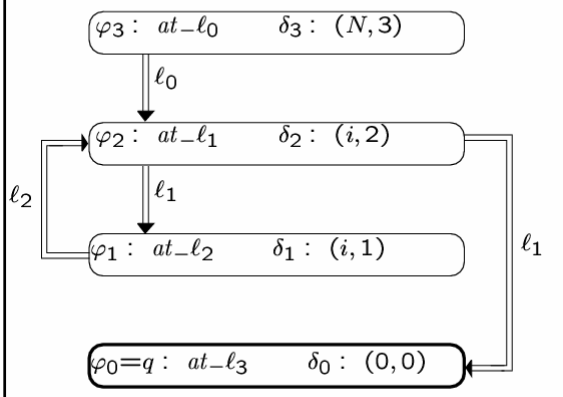
} for $i = 1, \ldots, m$

$$p \Rightarrow \Diamond q$$

complete !

Rank Diagrams

$\underbrace{at\_\ell_0}_{p} \Rightarrow \Diamond \underbrace{at\_\ell_3}_{q}$

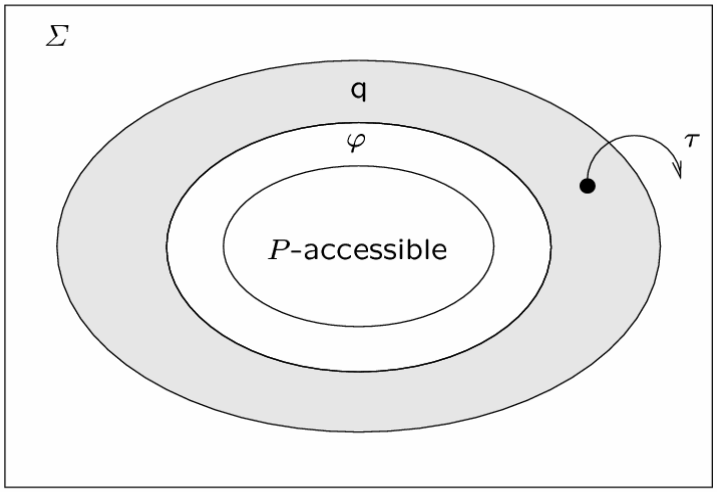


Nodes:
labeled by assertions and ranking functions

Well-formedness constraint:
Every node φ_i, $i > 0$, has a double edge departing from it.

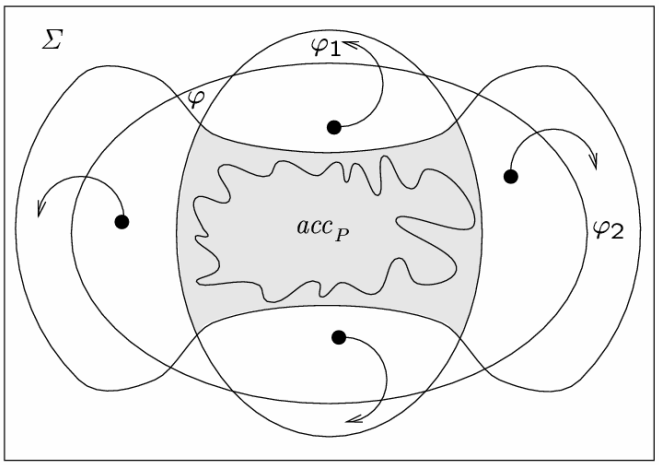
complete !

Strengthening



Find a stronger assertion φ that is inductive and implies the assertion q we want to prove.

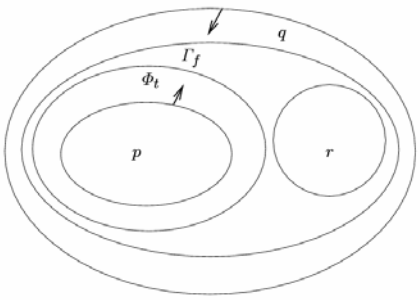
Completeness Proof (Rule INV)



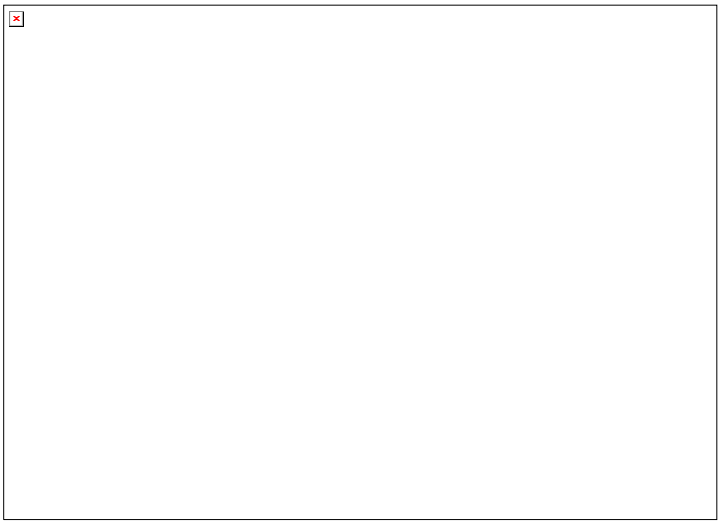
Although the assertion acc_P is inductive and strengthens any P-invariant, it is not very useful in practice.

Finding Inductive Assertions

1. Bottom-up methods:
   - Control invariants
   - Transition-validated invariants
   - Linear invariants
2. Top-down methods:
   - Forward propagation
   - Backward propagation



Model Checking



LTL model checking

- LTL is a logic for formalizing path-based properties
- Expansion law allows for rewriting until into local conditions and next
- LTL-formula φ can be transformed algorithmically into NBA $\mathcal{A}_\varphi$
  - this may cause an exponential blow up
  - algorithm: first construct a GNBA for φ; then transform it into an equivalent NBA
- LTL-formulae describe ω-regular LT-properties
  - but do not have the same expressivity as ω-regular languages

LTL model checking

- $S \models \varphi$ can be solved by a nested depth-first search in $S \otimes \mathcal{A}_{\neg\varphi}$
  - time complexity of the LTL model-checking algorithm is linear in S and exponential in $|\varphi|$
- Fairness assumptions can be described by LTL-formulae

  the model-checking problem for LTL with fairness is reducible to the standard LTL model-checking problem
- The LTL-model checking problem is PSPACE-complete
- Satisfiability and validity of LTL amounts to NBA emptiness-check

Summary: Core Techniques

- Deductive Verification
  - Prove invariance properties (manually find inductive assertions and apply rule)
  - Prove precedence properties (manually find inductive assertions and apply rule)
  - Prove response properties (manually find assertions and ranking function and apply rule)
  - Generate bottom-up invariants (e.g., linear invariants)
- Model Checking
  - LTL -> GNBA
  - GNBA -> NBA
  - Various constructions on GNBAs and NBAs
  - Determine if language of NBA is empty

Midterm on Thursday

- 20.12.2007, 2:00pm - 4:00pm in HS 002 Building E1 3
- closed-book except for a single A4 sheet of paper with your own notes. (You may use both sides of the sheet.)
- bring your ID

- **Requirement for admission to backup exam:** passing grade in either midterm or final (but not both)

- Some sample solutions are password protected. (Send us email to get the password.)