Richer Interface Automata

with Optimistic and Pessimistic Compatibility

Gerald Lüttgen
Software Technologies Research Group
University of Bamberg, Germany

(Joint work with Walter Vogler, University of Augsburg, Germany)

Interface Theories

- Traditional interfaces *(from 1960s)*
  - Deal with *types of data and operations*
  - Are implemented in most compilers
- Sequential, object-oriented interfaces *(from mid 1980s)*
  - *Behavioural types*, as supported by automated verification tools
  - E.g., pre-/post-conditions and invariants of classes and methods
  - Prominent example: **Meyer's contracts in Eiffel**
- Concurrent, behavioural interfaces *(from early 2000s)*
  - *Take concurrency into account*
  - Describe the interaction between system components
  - Prominent prototype: **de Alfaro/Henzinger's Interface Automata (IA)**

Semantic Framework

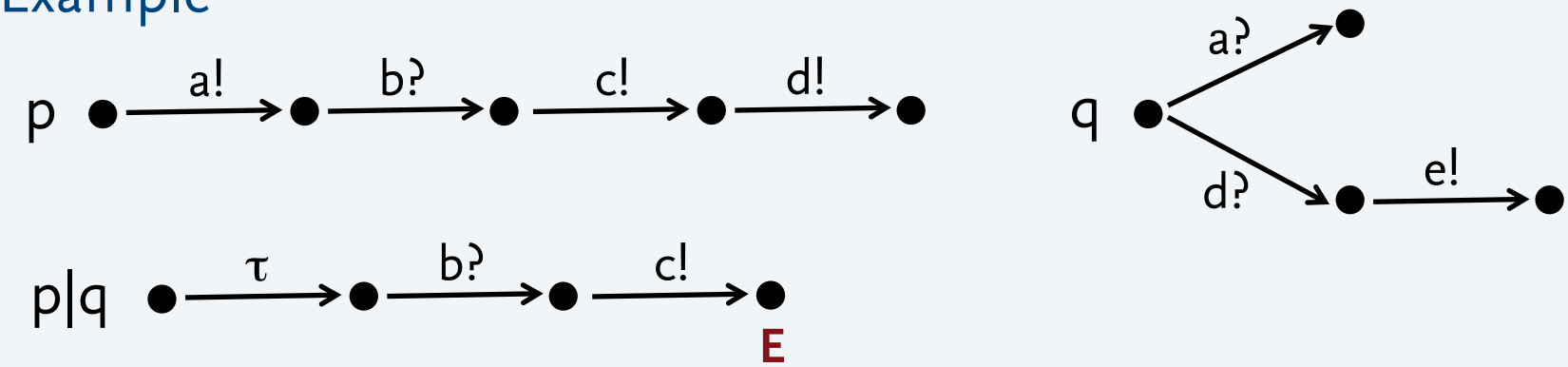
- **Labelled transition systems** representing component behaviour
  - *States* p connected via *transitions* p‘ – a –> p‘‘
  - Transitions labelled with *input* (i?), *output* (o!) or *internal* (τ) *actions*, taken from some alphabets

- Desiderata
  - **Parallel composition**
    - p|q with a sensible notion of *component compatibility*
  - **Compositional refinement preorder**
    - $p \leq q$ implies $p|r \leq q|r$ *(monotonicity of parallel composition)*
  - **Conjunction**
    - Supporting perspective-based specification
    - $r \leq p \wedge q$ if and only if $r \leq p$ and $r \leq q$ *(greatest lower bound)*
  - **Alphabet extension**
    - Permitting the addition of new `features´ when refining interfaces

Parallel Composition on IA

- Basic idea adopted from process algebra *(cf. Milner's CCS)*
  - *Synchronize on shared actions (a!, a?), resulting in the internal action τ*
  - Interleave on non-shared actions

- Example



- Optimistic notion of compatibility [Dill, late 1980s]
  - *Inputs can be delayed, outputs are immediate*
  - If output of p is not expected by q, then p|q is an **error state**
  - Remove states that can reach an error locally (i.e., via outputs or τs)

IA-Refinement

- An alternating simulation: the largest relation $\leq$ on states s. t. for all $p \leq q$
  - $q - a? \rightarrow q'$ implies $\exists\, p'.\ p - a? \rightarrow p'$ and $p' \leq q'$
  - $p - a! \rightarrow p'$ implies $\exists\, q'.\ q\ (- \tau \rightarrow)^* - a! \rightarrow q'$ and $p' \leq q'$

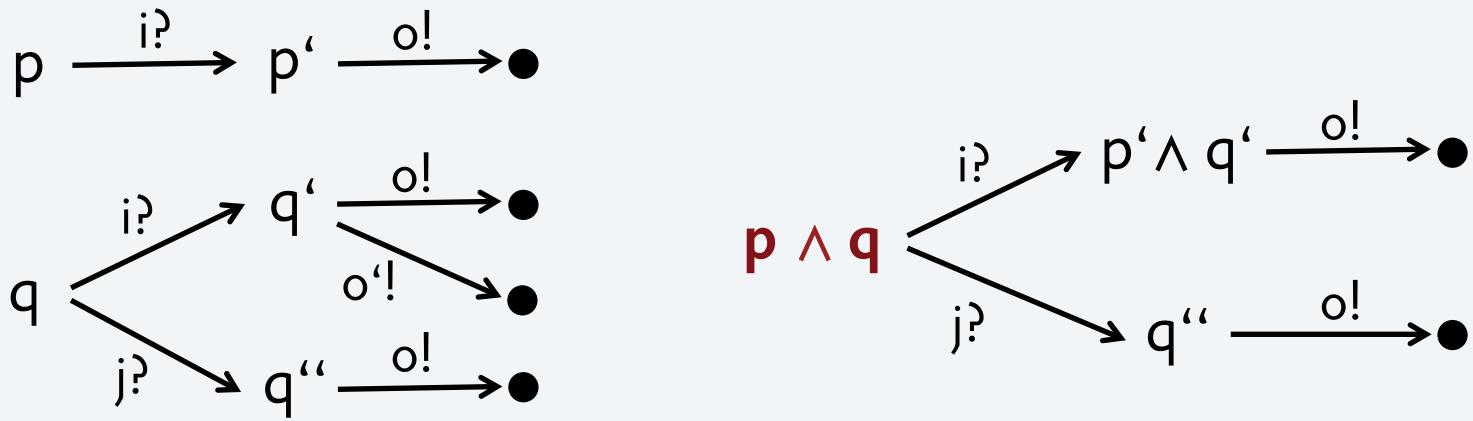
    (and similarly for τ-actions)

- Intuition
  - *Specified inputs* (outputs) *must be implemented* (are not required)
  - *Unspecified inputs* (outputs) *are always allowed* (are forbidden)

- Compositionality result for input-deterministic IA
  - **$p \leq q$ implies $p|r \leq q|r$ for any input-deterministic r**
  - Direct matching of inputs (i.e., without leading τs) is important

Conjunction on IA

- Example with interfaces that have the same alphabets



- Any set of interfaces is implementable (no inconsistencies)

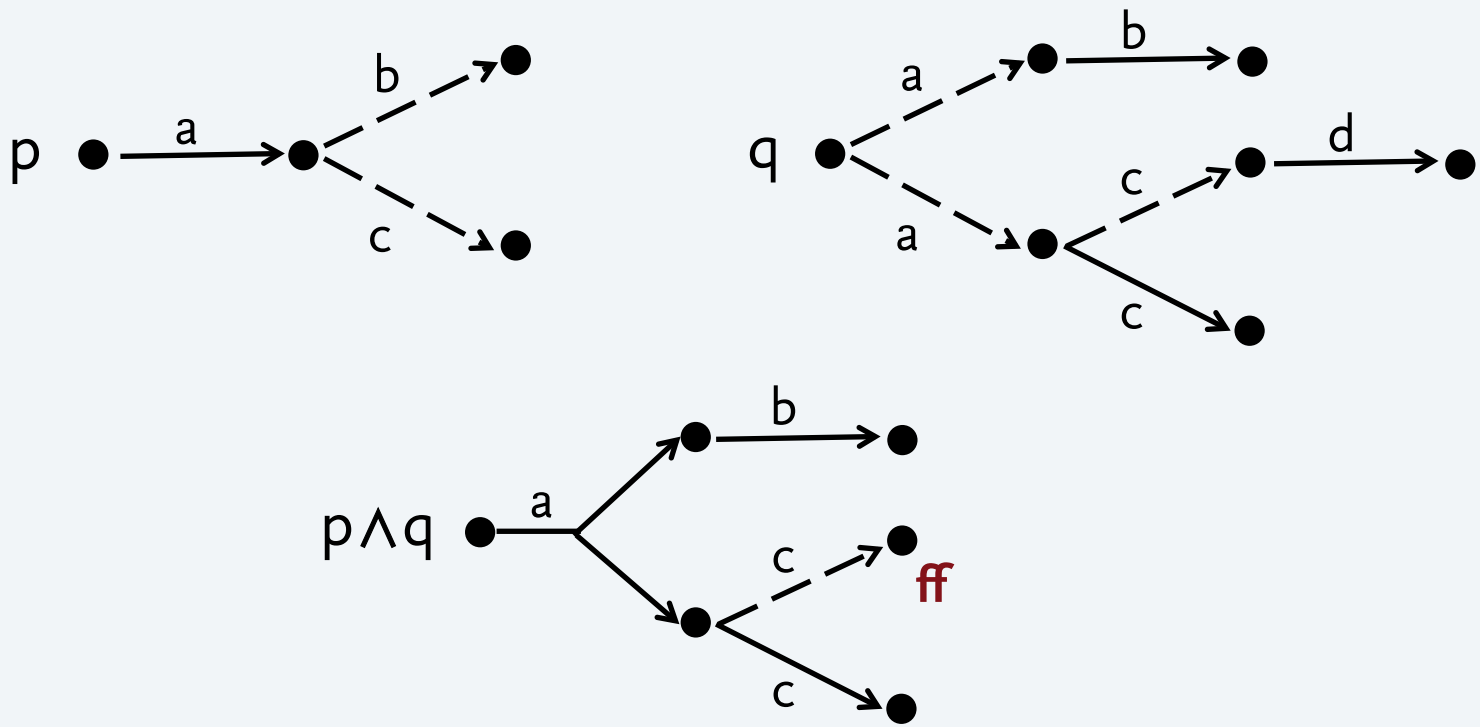
- **But BlackHole also means that IA cannot enforce outputs !!!**

I/O Modal Transition Systems (IOMTS)

- Adopt idea of modal transition systems (MTS) [Larsen, 1990s]
  - *Add a must-/may-modality to each action-labelled transition*
    - Must- (may-)transitions must (may) be implemented
    - Unspecified transitions are prohibited in any implementation/refinement
  - *Syntactic consistency, i.e., every must-transition is also a may-transition*
- IOMTS devised by Larsen, Nyman and Wasowski [ESOP'07]
  - MTS with input and output, as well as classic modal-refinement
  - Parallel composition and compatibility as in IA
  - **But: modal-refinement is not a precongruence for parallel composition**
- Modal-refinement is again an alternating simulation
  - Must-transitions in (IO)MTS are treated as input-transitions in IA
  - May-transitions in (IO)MTS are treated as output-transitions in IA
  - **But: τ-transitions on the specification side are ignored in IOMTS**

Conjunction on MTS *(no IO)*

- Example (cf. Benes, Cerna and Kretinsky [ATVA'11])



  - After `upper´ action c, something impossible (action d) must be done
  - **Inconsistencies are pruned** similar as in parallel composition for IA

- *Conjunction in the presence of τ-transitions is treated in [L., V. , 2012]*

Modal Interface Automata (MIA)

- Inherits from MTS
  - **Disjunctive must-transitions incl. τ-must-transitions, unlike in IOMTS**
- Inherits from IA
  - *Parallel composition*, including the notion of compatibility
  - *Input-determinism*, but relaxed due to disjunctive input-must-transitions
- Merges ideas from MTS and IA
  - *Refinement* as in MTS, but inputs are always allowed as in IA
    - All specified inputs are must-inputs
    - Parallel composition is monotonic, unlike in IOMTS
    - **Must-τs on specification side are considered, unlike in IOMTS**
  - *Conjunction*, where inputs (outputs) are treated as in IA (MTS)
- **Adds alphabet extension**
  - As advocated by Raclet, Caillaud et al. [Fund. Inform., 2011]

MIA-Refinement

- Refinement preorder
  - **Considers τ-must-transitions on the specification side**
  - *Permits alphabet extension on the implementation side*
  - *Another contribution:* definition of weak disjunctive must-transitions

- MIA-refinement ≤

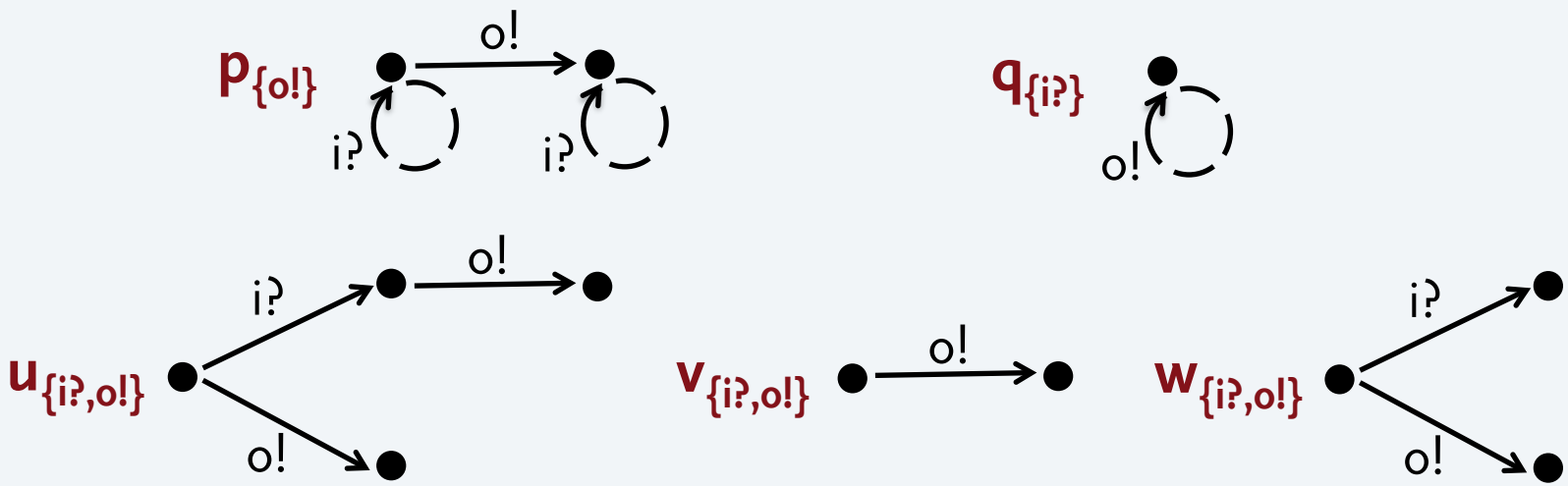
  Given MIAs P, Q s.t. $I_P \supseteq I_Q$, $O_P \supseteq O_Q$ and $p \leq q$

  1. $q - i \rightarrow Q'$ implies $\exists P'.\ p - i \rightarrow P'$ and $\forall p' \in P'\ \exists q' \in Q'.\ p' \leq q'$
  2. $q - o \rightarrow Q'$ implies $\exists P'.\ p = o \Rightarrow P'$ and $\forall p' \in P'\ \exists q' \in Q'.\ p' \leq q'$
  3. **$q - \tau \rightarrow Q'$ implies $\exists P'.\ p = \varepsilon \Rightarrow P'$ and $\forall p' \in P'\ \exists q' \in Q'.\ p' \leq q'$**

  4. $p - i \rightarrow P'$ and $i \notin I_Q$ implies $\forall p' \in P'.\ p' \leq q$
  5. $p \cdots o \cdots\rightarrow p'$ and $o \in O_Q$ implies $\exists q'.\ q \vdots\ o \vdots\rightarrow q'$ and $p' \leq q'$
  6. $p \cdots o \cdots\rightarrow p'$ and $o \notin O_Q$ implies $\exists q'.\ q \vdots\ \varepsilon \vdots\rightarrow q'$ and $p' \leq q'$
  7. $p \cdots \tau \cdots\rightarrow p'$ implies $\exists q'.\ q \vdots\ \varepsilon \vdots\rightarrow q'$ and $p' \leq q'$

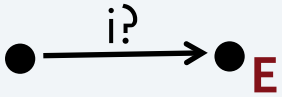
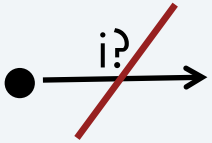
Conjunction on MIA

- Conjunction for interfaces with the same alphabet
  - Treatment of inputs as for IA and of outputs and τs as for MTS
- But problem with conjunction for dissimilar alphabets



- *Extending alphabets morally means to add `neutral´ may-loops*
  - Here, **p∧q** should intuitively be implementable by **u** and **v** but not **w**
  - **No MIA** has **u** and **v** but not **w** as implementations, since it cannot have an initial i?-must-transition because of **v** !!!

The Pessimistic Approach

- Restrictive, pessimistic notion of compatibility
  - Advocated by Bauer, Hennicker et al. in their MIO theory [TACAS'10]
  - *If any error state in a parallel composition can be reached via **any** actions (including input actions), then the composition is not defined*
  - This approach distinguishes between  and 

- Trade-off
  - *Classic view of may-inputs* (i.e., input allowed or prohibited)
  - **Much fewer parallel compositions are permitted**
- Investigating MIA for pessimistic compatibility
  - *Here, one can do away with input-determinism and with the requirement that inputs are must-inputs*
  - `Standard´ modal-refinement, with compositional parallel operator

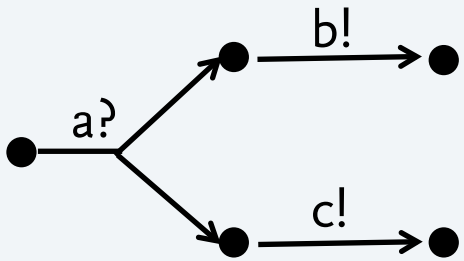
Conjunction with Alphabet Extension

- *Conjuncts with dissimilar alphabets*

- *Alphabet extension* **$[p]_{\{c!\}}$** *of conjunct p*

- *Conjunctive composition*

$$p \wedge q = [p]_{\{c!\}} \wedge [q]_{\{\}}$$



Conclusion: Richer Interface Automata

- **MIA is a compositional interface theory that allows one to**
  - Enforce outputs
  - Express disjunctive must-transitions
  - Specify non-deterministic behaviour (unlike Raclet et al. [Fund. Inform., 2011])
  - Abstract from internal computation
  - Compose interfaces in parallel, conjunctively and disjunctively
  - Interpret compatibility optimistically or pessimistically
  - Extend alphabets while refining

- Some questions left to future work
  - Are there `clean´ interface theories in-between the optimistic and pessimistic approaches?
  - *Is there a possibility to allow may-inputs while maintaining a true open-systems view?*

Selected Literature

- L. de Alfaro, T. Henzinger. *Interface-based design*. In Engineering Theories of Software-Intensive Systems. NATO Science Series 195. Springer, 2005.
- S. Bauer, P. Mayer, A. Schroeder, R. Hennicker. *On weak modal compatibility, refinement, and the MIO Workbench*. In TACAS'10, LNCS 6015, pp. 175-189. Springer, 2010.
- K. Larsen, U. Nyman, A. Wasowski. *Modal I/O automata for interface and product line theories*. In ESOP'07. LNCS 4421, pp. 64-79. Springer, 2007.
- G. Lüttgen, W. Vogler. *Modal Interface Automata*. In TCS'12. LNCS 7604, pp. 265-279. Springer, 2012. Extended version appeared in LMCS 9(3:4), August 2013.
- J. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay, R. Passerone. *A modal interface theory for component-based design*. Fund. Inform. 107:1-32, 2011.