Universität Paderborn

UNIVERSITÄT DORTMUND

A Formal Approach to Service Specification and Matching based on Graph Transformation

Reiko Heckel
(Dortmund / Paderborn)
Alexey Cherchago and **Marc Lohmann**
(Paderborn)

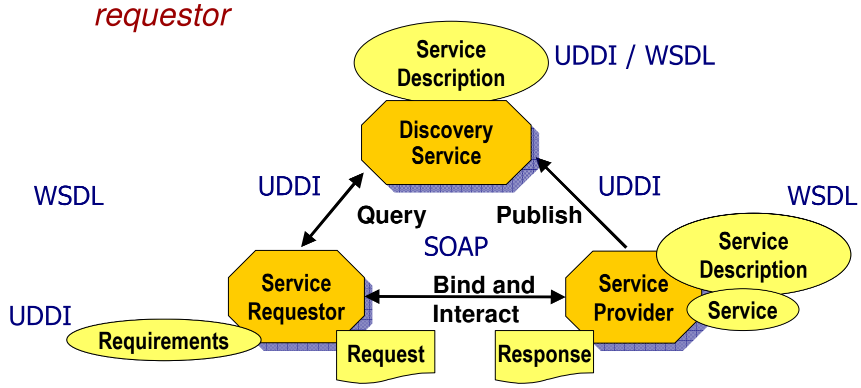
WSFM@Concur2004, Pisa
Feb 22-23, 2004

1

UNIVERSITÄT DORTMUND

Service-Oriented Architectures

A Web service is a *component* deployed on a *Web accessible platform* provided by a *service provider* to be *discovered* and *invoked* over the Web by a *service requestor*



Not enough to allow dynamic discovery and binding!

2

Example: Car Rental Service

<<interface>>
RentalServiceRequired
reservCar(c:Customer, car:Car, ri:RentalInfo)
...

<<interface>>
RentalServiceProvided
makeReserv(c:Customer, car:Car, ri:RentalInfo): EContract
...

Matching *provider* and *requestor* specification within *registry* must ensure compatibility of

- Data types
  - Does Customer have the same meaning for requestor and provider?
- Operation signatures
  - Can provider operation be supplied with suitable parameters from a call of requestor operation?
- Behavior
  - Does provided operation actually carry out what is expected by a requestor?

Data Types and Signatures

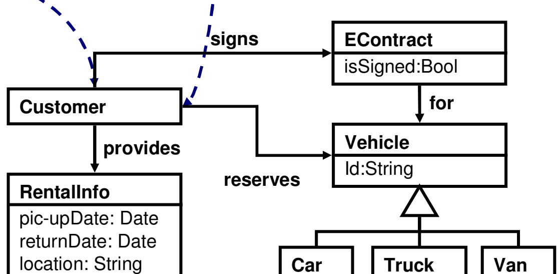
<<interface>>
RentalServiceRequired
reservCar(c:Customer, car:Car, ri:RentalInfo)
...

<<interface>>
RentalServiceProvided
makeReserv(c:Customer, car:Car, ri:RentalInfo): EContract
...

- Reorder and rename pars
- Skip input of requestor
- Ignore output of provider

Data types: parties use common domain model (ontology)

Operation signatures: Zaremski and Wing: *Signature matching: A tool for using software libraries.* TOSEM 1995.



Behavior: Operation Contracts

Pre-condition:
Customer provides rental info and chooses car

Effect:
Car is reserved for customer

Required

- Formal specification (logic, graph transformation, …) for automatic matching
- Integration into mainstream SW development methods (UML) for wider applicability

Outline

- Contracts as graph transformation rules
- Semantics of rules
- Semantic / syntactic compatibility, soundness

5

UNIVERSITÄT DORTMUND

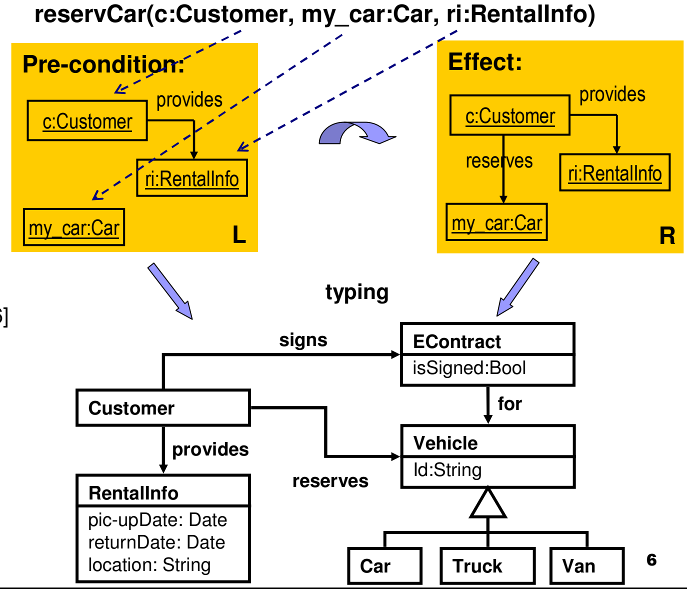
Contracts as Graph Transformation Rules

Signature: **reservCar(c:Customer, my_car:Car, ri:RentalInfo)**

Behavior:
GT rule

Typed DPO
[Corradini et al 96]

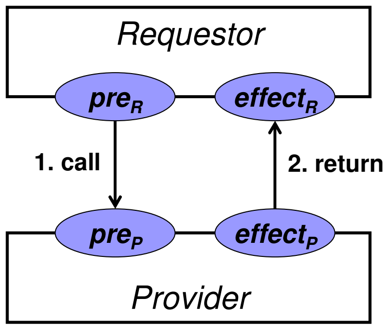
Data types:
type graph



6

UNIVERSITÄT DORTMUND

What is the right notion of compatibilty? That depends on …



how services should interact:

1. Requestor guarantees ***pre$_R$***
   → Provider assumes ***pre$_P$***
2. Provider guarantees ***effect$_P$***
   → Requestor assumes ***effect$_R$***

… a *contravariant* relation.

what it should mean, that:

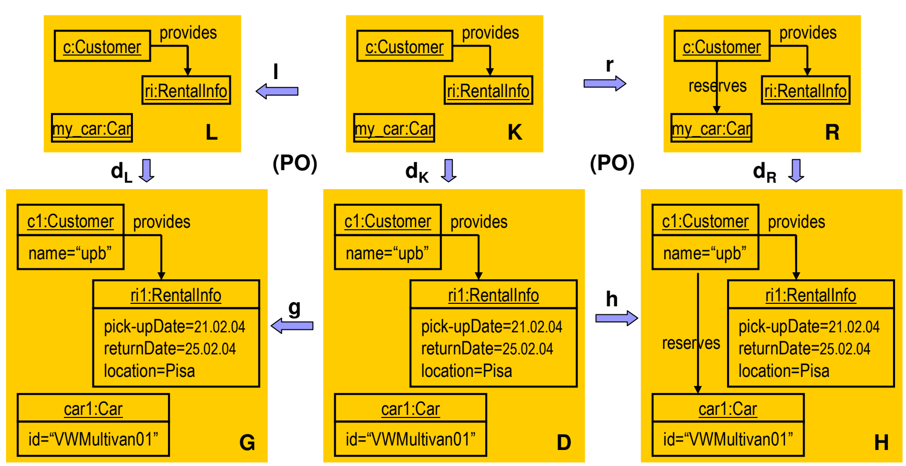
- an *assumption* is correct
- a *guarantee* is fulfilled

… a question about the *semantics of contracts.*

7

UNIVERSITÄT DORTMUND

Operational Semantics: The DPO Approach



- *L* is embedded into graph ***G.***
- The elements of ***G*** matched by ***L- l(K)*** are removed.
- The elements matched by ***R - r(K)*** are added to *D.*

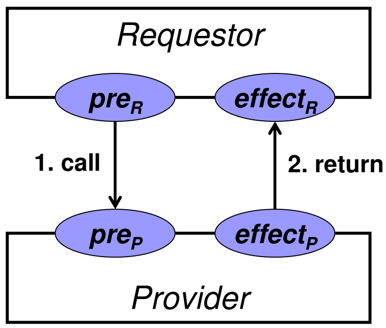
The changes to *G* are exactly those specified by the rule

8

UNIVERSITÄT DORTMUND

Loose Semantics of Contracts

Requestor has only loose idea of behavior of the other service



Provider has complete info, but may prefer not to publish everything

→ Contracts are incomplete specifications of service behavior

$$\begin{tikzcd} L \arrow[d, "d_L"'] \arrow[dr, phantom, "(PB)"] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "d_K"] \arrow[dr, phantom, "(PB)"] & R \arrow[d, "d_R"] \\ G & D \arrow[l, "g"] \arrow[r, "h"'] & H \end{tikzcd}$$

Formally: Double-Pullback (DPB), allows unspecified

- Deletion: *at least* elements of G matched by L - l(K) are removed
- Creation: *at least* elements matched by R - r(K) are added to D

(faithful) transition vs. transformation

9

UNIVERSITÄT DORTMUND

Contracts as Rules, revisited

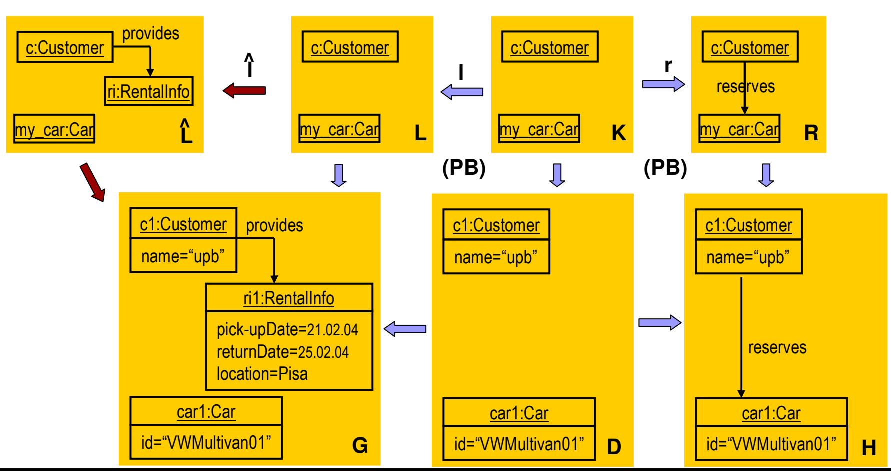
→ Positive Application Conditions

Precondition: what must be

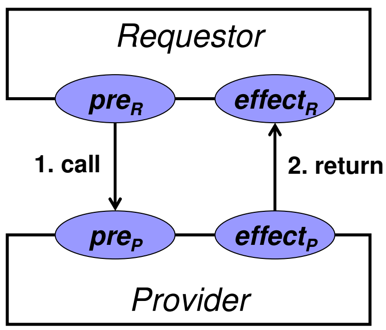
- present before, no matter what happens later

Effect: what must be

- deleted
- preserved
- created



What is the right notion of compatibility? That depends on …



how services should interact:

1. Requestor guarantees $\boldsymbol{pre_R}$
   → Provider assumes $\boldsymbol{pre_P}$
2. Provider guarantees $\boldsymbol{effect_P}$
   → Requestor assumes $\boldsymbol{effect_R}$

… a *contravariant* relation.

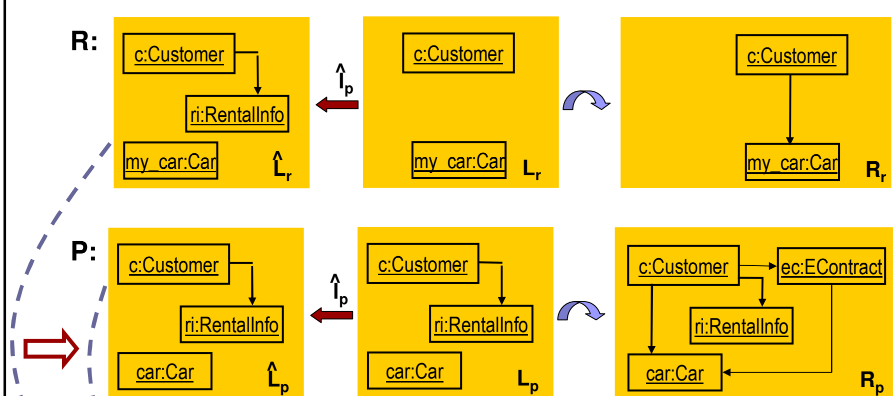
what it should mean, that:

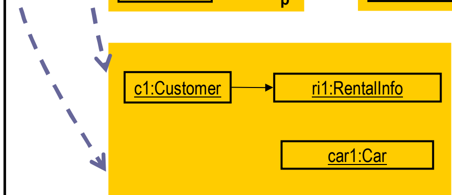
- an *assumption* is correct
- a *guarantee* is fulfilled ✓

… a question about the *semantics of contracts.*

11

Semantic Compatibility



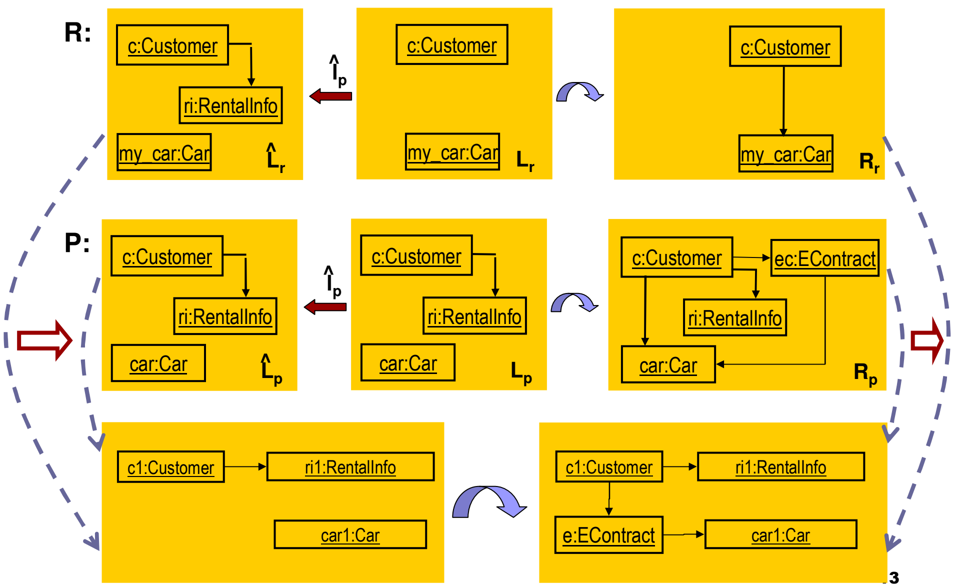


1. $\boldsymbol{pre_R \rightarrow pre_P}$ **:** applicability of requestor rule **implies** applicability of provider rule
2. $\boldsymbol{effect_P \rightarrow effect_R}$ **:** transition via provider rule **implies** transition via requestor rule.

12

UNIVERSITÄT DORTMUND

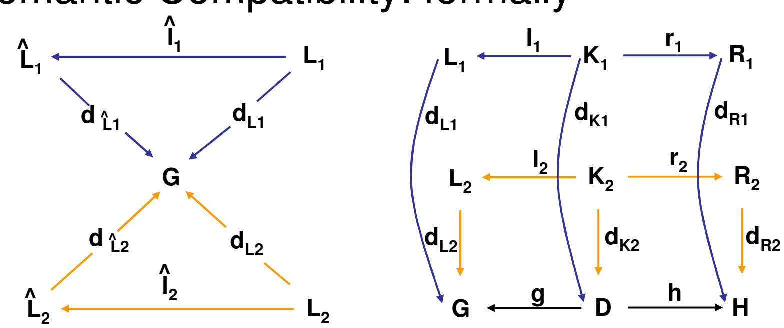
Semantic Compatibility



13

UNIVERSITÄT DORTMUND

Semantic Compatibility: formally



$(p_1, \hat{L}_1)$ semantically matches $(p_2, \hat{L}_2)$, in symbols $(p_2, \hat{L}_2) \models_{match} (p_1, \hat{L}_1)$, iff:

(i) for all graphs G, if there exists $d_{\hat{L}_1} : \hat{L}_1 \to G$ s.t. $d_{L_1} := d_{\hat{L}_1} \circ \hat{l}_1$ satisfies IC of p_1, then there exists $d_{\hat{L}_2} : \hat{L}_2 \to G$ s.t. $d_{L_2} := d_{\hat{L}_2} \circ \hat{l}_1$ satisfies IC of p_2, and

(ii) for all spans $t : (G \xleftarrow{g} D \xrightarrow{h} H)$, if there exists a transition $G \overset{p_2/d_2}{\rightsquigarrow} H$, then there exists a transition $G \overset{p_1/d_1}{\rightsquigarrow} H$ using the same bottom span t.

14

What do we have?

Semantic compatibility relation |= over rules

- quantified over sets of all graphs and transitions
- cannot be verified directly

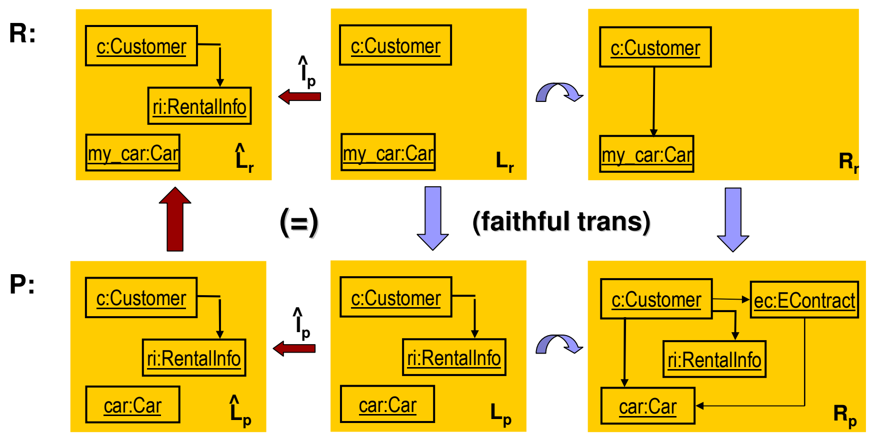
Goal: syntactic matching relation |-- over rules such that

- Soundness: p_2 |-- p_1 *implies* p_2 |= p_1
- Completeness: p_2 |= p_1 *implies* p_2 |-- p_1

15

UNIVERSITÄT DORTMUND

Syntactic Matching Relation



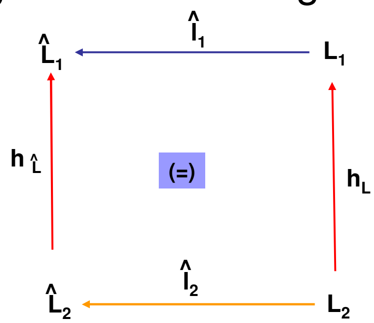
***pre$_R$* → *pre$_P$* :** requestor must provide all information necessary for the execution of the provider operation,

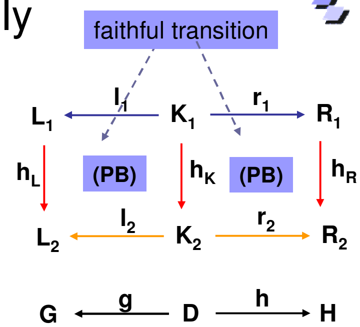
***effect$_P$* → *effect$_R$* :** effect of the provided operation must include those expected by the requestor.

16

UNIVERSITÄT DORTMUND

Syntactic Matching: formally





$(p_1, \hat{L}_1)$ syntactically matches $(p_2, \hat{L}_2)$, in symbols $(p_2 : s_2, \hat{L}_2) \vdash_{match} (p_1 : s_1, \hat{L}_1)$, iff:

(i) there exists an injective graph homomorphism $h_{\hat{L}} : \hat{L}_2 \to \hat{L}_1$ s.t. $h_{\hat{L}} \circ \hat{l}_2$ satisfies IC of p_2, and

(ii) there exist graph homomorphisms $h_L : L_1 \to L_2$, $h_K : K_1 \to K_2$, and $h_R : R_1 \to R_2$ s.t. the diagrams (a), (b), and the diagram on the left commute, and the diagrams (a) and (b) represent a faithful transition.

17

UNIVERSITÄT DORTMUND

What do we have?

Semantic compatibility relation |= over rules

- quantified over sets of all graphs and transitions
- cannot be verified directly

Goal: syntactic matching relation |-- over rules such that

- ✓ Soundness: ***p_2 |-- p_1** implies **p_2 |= p_1***
- ☐ Completeness: ***p_2 |= p_1** implies **p_2 |-- p_1***

18

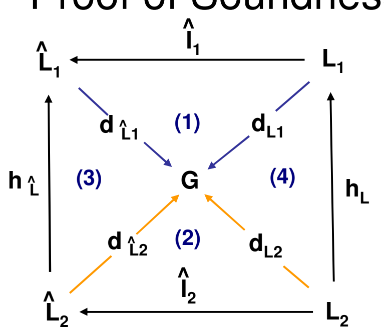
Summary & Future Work

- Formal approach to service specification matching.
- Operation contracts are GT-rules with loose semantics.
- Semantic and syntactic matching relations.
- Soundness of matching.

- Refinement of semantic compatibility (→ completeness of syntactic matching).
- Extension to typed graphs with attributes and subtyping.
- Logic / XML-representation of contracts: RDF in DAML-S
- Tool support for computing syntactic matching based on RDF graph matching with RDQL

UNIVERSITÄT DORTMUND

Proof of Soundness



$$\begin{tikzcd}
L_1 \arrow[d, "h_L"'] \arrow[dr, phantom, "(a)"] & K_1 \arrow[l, "l_1"'] \arrow[r, "r_1"] \arrow[d, "h_K"] \arrow[dr, phantom, "(b)"] & R_1 \arrow[d, "h_R"] \\
L_2 & K_2 \arrow[l, "l_2"'] \arrow[r, "r_2"] & R_2
\end{tikzcd}$$

$$G \xleftarrow{g} D \xrightarrow{h} H$$

To prove: $\hat{p}_2 \vdash_{match} \hat{p}_1$ *implies* $\hat{p}_2 \models_{match} \hat{p}_1$

(i) for all graphs G, if there exists $d_{\hat{L}_1} : \hat{L}_1 \to G$ s.t. $d_{L_1} := d_{\hat{L}_1} \circ \hat{l}_1$ satisfies IC of p_1, then there exists $d_{\hat{L}_2} : \hat{L}_2 \to G$ s.t. $d_{L_2} := d_{\hat{L}_2} \circ \hat{l}_1$ satisfies IC of p_2

$d_{\hat{L}_2} : \hat{L}_2 \to G$ is $d_{\hat{L}_1} \circ h_{\hat{L}}$ (diagram (3) commutes).

$d_{L_2} = d_{\hat{L}_2} \circ \hat{l}_2$ satisfies IC of p_2 because of this commutativity and the fact that $h_{\hat{L}} \circ \hat{l}_2$ satisfies IC of p_2.

Proof of Soundness (cntd.)

$$
\begin{tikzcd}
\hat{L}_1 \arrow[dr, "d_{\hat{L}1}"'] & \arrow[d, phantom, "(1)"] & L_1 \arrow[ll, "\hat{l}_1"'] \arrow[dl, "d_{L1}"] \\
\arrow[r, phantom, "(3)"] & G & \arrow[l, phantom, "(4)"] \\
\hat{L}_2 \arrow[uu, "h_{\hat{L}}"] \arrow[ur, "d_{\hat{L}2}"] & \arrow[u, phantom, "(2)"] & L_2 \arrow[ll, "\hat{l}_2"] \arrow[uu, "h_L"'] \arrow[ul, "d_{L2}"']
\end{tikzcd}
\qquad
\begin{tikzcd}
L_1 \arrow[d, "h_L"] \arrow[dd, bend right, "d_{L1}"'] & K_1 \arrow[l, "l_1"'] \arrow[r, "r_1"] \arrow[d, "h_K"] \arrow[dd, bend right, "d_{K1}"'] & R_1 \arrow[d, "h_L"] \arrow[dd, bend right, "d_{R1}"'] \\
L_2 \arrow[d, "d_{L2}"] & K_2 \arrow[l, "l_2"'] \arrow[r, "r_2"] \arrow[d, "d_{K2}"] & R_2 \arrow[d, "d_{R2}"] \\
G & D \arrow[l, "g"'] \arrow[r, "h"] & H
\end{tikzcd}
$$

(ii) for all spans $t : (G \xleftarrow{g} D \xrightarrow{h} H)$, if there exists a transition $G \overset{p_2/d_2}{\rightsquigarrow} H$, then there exists a transition $G \overset{p_1/d_1}{\rightsquigarrow} H$ using the same bottom span t.

Both transitions can be vertically composed using the composition of the underlying pullback squares.

Faithfulness of the composed transition follows from the fact that IC of d_{L_1} follows from that of h_L and d_{L_2} (analogously for the right-hand side).