The Logic of Distributed Systems

- Jon Barwise and Jerry Seligman developed a formal framework to model *information flow*. (The book *Information Flow* appeared in 1997.)
- They call it "Logic of Distributed Systems".
- It involves something like *constraints*, but the formal tools are more abstract than situation theory it´s said to have started from. There are no *situations* in it!

A Note on Our Presentation

- Although all topics are simplified to their core in this course, this applies in the extreme to Barwise and Seligman´s theory of information flow. Most details, refinements, some of their ideas on logic and all metalogical properties are left out!
- The aim of these slides is to give you an idea of their methods and basic proposals.

Methodological Starting Points

- The analysis should be a *conceptual analysis* or a formal *construction* of some important conceptual features of information flow (or *some* concept thereof).
- It is based on principles deemed to be correct. The set of them might be incomplete.
- It is *not* an analysis and reconstruction of our present day intuitions.

Methodological Starting Points (II)

- Examples show modelling real information flow (IF) requires many disciplines (logic, cognitive sciences, sociology…).
  A model of IF *in its own terms* tries to arrive at (abstract) *laws of information flow*.
- The model in its generality covers both physical systems and mathematical proofs!

Centre for the Study of Logic, Language, and Information

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Manuel Bremer, Daniel Cohnitz
Information Flow and Situation Semantics
ESSLLI 2002

Distributed Systems

- IF depends on relationships in *a distributed system* (e.g., a telephone connection).
- *How* the system is carved up is part of the model of the IF to be explained.
- The parts of a distributed system are related to each other *by the system* as a whole.
- Regularities ensure the uniform behaviour of the system. – Has it to be deterministic?

Distributed Systems Diagram

You can see a flashlight as a distributed system in which the parts are connected by being parts of the same system.

(This diagram tells us nothing about information flow, so far.)

Determined Distributed Systems

- If you look at examples: the more random a system is the less information will flow.
  (You cannot predict what will happen next in some other part of it.)
- IF depends on non-accidental connections, a "spurious regularity" is not enough. IF depends on a *reliable* process. This is a *necessary condition* of IF.
- Note the connection between IF and *non*-randomness (somewhat contrary to the Shannon approach).

Epistemological Principles

- IF is the missing *link* between justified belief and knowledge. IF-Theory is related to epistemology, therefore.
- IF-Theory follows Dretske in his externalist epistemology: IF is a factual concept, not depending on us knowing some conditions being met. Information *just flows*.
- IF complies, e.g., with the Xerox-Principle.

Information Flow and Causality

- IF is required to be reliable. That doesn´t mean that is –as might be *reliability*– can be analysed using the concept of causality.
- The direction of IF is not necessarily aligned with the direction of causation. Informational dependence isn`t causal.
- In case of loose connections a causal connection might be not sufficient for IF.

Carrying Information

- A lot of things are said to carry information:
  (1) The rifle shot carried the information that the king was dead to the whole city. [The rifle has the property *being fired*]
  (2) The e-mail message bore the information that Albert would be late for dinner. [The message has the property *containing the words…*]
- A common denominator is that in all cases some object having some property matters:
  a´s being F carries the information that b is G

Basic Idea

- Parts of a distributed systems are particulars with properties, they are of some *type*.
- Types are connected by *constraints*.
- The regularities of the system are represented by the constraints of the complete model of that system as a whole.
- Information flows along these constraints at the system level using some *local logic*.

Information Content (Outline)

For a cognitve system x with prior knowledge K a part *a* of a distributed system *being F* carries the information that another part *b* of that system is *G* if x could legitimately infer from the constraints of the distributed system, given some local logic, that *b is G* from *a is F* together with x´s prior knowledge K (but not from K alone).

Centre for the Study of Logic, Language, and Information

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Manuel Bremer, Daniel Cohnitz
Information Flow and Situation Semantics
ESSLLI 2002

Classifications

- A classification is a structure $A=\langle U,\Sigma_A,|=_A\rangle$ where U is the set of objects to be classified (the *tokens* of A), Σ_A the set of objects used to classify the tokens (the *types* of A), and $|=_A$ is a binary relation between U and Σ_A determining which tokens are of which type
- $a |=_A F$ says that object a is of the type F

  You can think of a as a situation and F as an infon supported by a.
  (Hey – but why then not say "F(a)" and use First Order Logic?)

Illustrating Classifications

- For the purpose of later diagrams we illustrate a classification relation:

Sequents

- Constraints relate types; we first introduce a more general logical relation:
  Given a classification A a *sequent* is a pair $\langle \Gamma, \Lambda \rangle$ of sets of types of A.
- A token a *satisfies* the sequent $\langle \Gamma, \Lambda \rangle$ if $(\forall \alpha \in \Gamma)(a \models_A \alpha) \Rightarrow (\exists \delta \in \Lambda)(a \models_A \delta)$
- Γ *entails* Λ in A: $\Gamma \vdash_A \Lambda$
  if every token of A satisfies $\langle \Gamma, \Lambda \rangle$.

Constraints

- If $\Gamma \vdash_A \Lambda$ then the pair $\langle \Gamma, \Lambda \rangle$ is a *constraint* supported by the classification A.
  Note: In general a constraint is satisfied by a token if the token is of at least one of the types in Λ (Λ is taken disjunctively).
- The set of all constraints supported by A is the complete theory of A, Th(A).
- Note the following special cases:
  - Γ, Λ being singletons: $\Gamma \vdash_A \Lambda$ means that Γ logically entails Λ in A
  - $\Gamma \vdash_A$ with right side being empty means in A no token is of type Γ
  - $\Gamma, \Lambda \vdash_A$ means, accordingly, that Γ and Λ are mutually exclusive
  - $\vdash_A \Lambda, \Gamma$ means that every token of A is of least of one of the types

Information Channels (Outline)

- In the outline we said that information flows in a distributed system. The system can be considered an *information channel*.
- We also said that IF involves the reliable regularities/constraints that connect parts of the system with each other *via* the system.
- This requires mappings from parts to the system as a whole, called "infomorphisms".

Infomorphisms

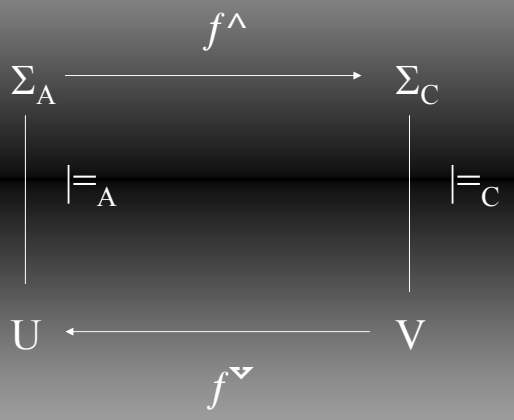
Let $A=\langle U,\Sigma_A,|=_A\rangle$ and $C=\langle V,\Sigma_C,|=_C\rangle$ be two classifications. An *infomorphism* between A and C is a pair $f=\langle f^\wedge, f^\vee\rangle$ of functions, such that for all tokens c of C and all types α of A the following (defining statement for informorphisms) is true:

$$f^\vee(c)|=_A \alpha \quad \text{iff} \quad c|=_C f^\wedge(\alpha)$$

Infomorphisms Described

- We call $f^{\vee}$ "f-up" and $f^{\wedge}$ "f-down". They two functions operate in opposite directions: f: A $\rightleftarrows$ C. f-up maps tokens from the "right-hand" classification to tokens of the other. f-down maps types the other way.
- Two formulas might express the relations:
  (1) Iff the target token (in A) is of some type the token (in C) is of the target type.
  (2) The type of the picture is the pictured type of the object.

Infomorphism Diagram



Centre for the Study of Logic, Language, and Information

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Manuel Bremer, Daniel Cohnitz
Information Flow and Situation Semantics
ESSLLI 2002

Infomorphism Example

- We are mainly interested in infomorphism that map parts (e.g. a switch) to the whole system (e.g. a circuit with a bulb).
- The classification concerning switches has a set of switches as objects and types that apply to switches. The classification of the whole distributed system contains switches *build into a circuit* and types *of these*.

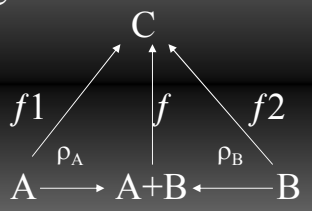
Infomorphism Example (II)

- If in our theory of (isolated) switches we have "a pressed switch shows red top", in our theory of flashlights (with switches) we have "in a flashlight a pressed switch shows red top". Consider now a switch build in.
- Expressed with the fundamental property:

$$\underline{\text{the switch}} \quad \underline{\text{shows red top}} \Leftrightarrow \underline{\text{the flashlight}} \quad \underline{\text{has red top shown}}$$
$$f^{\vee}(c) \quad \models_A \quad \alpha \qquad\qquad c \quad \models_C \quad f^{\wedge}(\alpha)$$

Infomorphisms Added Up

Infomorphisms can be added up to make the following diagram commute :



A+B is the sum of the classifications A and B, ρ_A being the projection from A onto A+B, so that $\rho_A^{\wedge}(\alpha) = \alpha_A$ (the A-copy of α), and on tokens $\rho_A^{\vee}((a,b))=a$. *f* being the composed infomorphism.

Regularities at System Level

- In classifications of parts constraints give us the theory of the (isolated) parts (e.g. bulbs)
- If parts come build a distributed system we not only have the sum of the theories of the parts, but regularities that govern the distributed system (e.g. a pressed switch *lights* the bulb).
- *These regularities* give us IF. For a system its classification C and Th(C) model them.

Regularities at System Level (II)

- The classification of switches A_1 contains a type *Pressed*, the classification of bulbs A_2 contains a type *On*, but they are not related.
- On the system level there are types *corresponding* to *Pressed* (say $Pressed_f$) and to *On* (say On_f) which at the system level C are connected by constraints: $Pressed_f \vdash_C On_f$
  (If a flashlight with pressed button has a bulb that is on, for example)

Information Channel Defined

- An information channel consists of an indexed family $\boldsymbol{C} = \{f_i: A_i \rightleftarrows C\}_{i \in I}$ of informorphisms with a common codomain C, the *core* of the channel.
- The intuition is: the A_i are the (classification of) parts of the distributed system C, and it is by virtue of being part of C that tokens the A_i carry information about one another.

Connected Tokens

- Two parts (tokens of constituent classifications) are *connected* if the same token of the co-domain is mapped onto them.
- Example:
  A switch s is connected to a bulb b if the infomorphism between the switch classification and the system classification, resp. the co-domain (i.e. the domain of flashlights) maps some flashlight token c to the switch s and the infomorphism between the bulb classification and the system classification maps the very same flashlight c to the bulb b. s and b, therefore, are the switch and bulb of the same flashlight. Information only flows in the context of a particular token of the co-domain

Information Flow (Outline)

Suppose A and B are constituent/part classifications in an information channel with core C. A token a of type α in A carries the information that a token b is of type δ in B relative to the channel ***C*** if a and b are connected in C and the translation of α entails the translation of δ in Th(C).

Information Flow (Example)

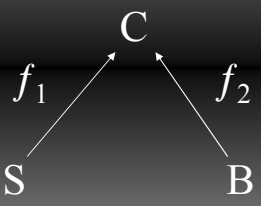
- Leaving aside some details –for later–, say:
  B is a bulb classification (with tokens b…), S is a switch classification, $f_1^{\vee}(c) = b$ for $c \in W$ (the domain of flashlights), $f_2^{\vee}(c) = s$ (i.e. b and s are connected in a flashlight) $f_1\hat{}(On) = On_f$ (On_f being a type of flashlights), and $f_2\hat{}(Press) = Press_f$ with the constraint holding
  $Press_f \mid\!\sim_C On_f$, i.e. $f_2\hat{}(Press) \mid\!\sim_C f_1\hat{}(On)$
- *Press(s)* carries the information *On(b)* since the corresponding types are connected at the system level. We have the information that the bulb is on *because* the switch is pressed.

Reasoning At a Distance (Idea)

- To get information about some part of a system having information about another part we have to *reason at a distance*, this means: We employ our knowledge about the distributed system and its regularities.
- In Detail:
  We see the part we know about as part of a system, apply some known regularities of the system to this representation of it, derive a system representation of the other part we are looking for, and, finally, translate this back into some simple information about that distant part.

Reasoning At a Distance (Diagram)

- Let C be the system, S and B be two constituent classifications with informorphisms



- We have to reason –some how– from S to C and then from C to B, along infomorphisms

f-Intro and f-Elim

- The rules concern mappings of types. Consider an infomorphism between A and C, we reason using these two rules:
- *f*-Intro: $\dfrac{\Gamma^{-f} \vdash_A \Delta^{-f}}{\Gamma \vdash_C \Delta}$ resp. $\dfrac{\Gamma \vdash_A \Delta}{\Gamma^{f} \vdash_C \Delta^{f}}$

(Γ^{f} the set of translations of Γ, Γ^{-f} the set whose translations are in Γ.)

- *f*-Elim: $\dfrac{\Gamma^{f} \vdash_C \Delta^{f}}{\Gamma \vdash_A \Delta}$

f-Intro

- *f*-Intro says we can take a *valid* A-sequent into a *valid* C-sequent. (so map the types)
- *f*-Intro *preserver validity*: If c were a counterexample to $\Gamma \mid\!\sim_C \Delta$ $f(c)=a$ would be a counterexample to $\Gamma^{-f} \mid\!\sim_A \Delta^{-f}$.
- Example: If it is valid in A that pressed switches show a red top, it is valid in C that flashlights with their switches pressed have a red top of their switch shown.
- *f*-Intro *doesn`t* preserve *non-validity*. Since some constraints *start* with the system level.

f-Elim

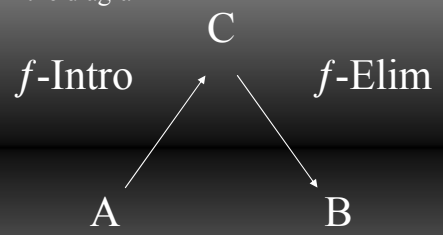
- *f*-Elim says we can take a *valid* C-sequent into a *valid* A-sequent. (so map the types)
- *f*-Elim *doesn`t preserve validity*: There can be a valid constraint $\Gamma^f \mid\!\sim_C \Delta^f$ but $\Gamma \mid\!\sim_A \Delta$ has a counterexample, although a counter-example cannot be $f(c)$ for any c.
- Example: Flashlights c with pressed switches make light, but some switches a –those not build in– don´t (but they aren´t connected either).
- *f*-Elim *preserves non-validity*.

f-Intro and *f*-Elim (Use)

- The validity preserving nature of the *f*-Intro rule tells us that any constraint that holds for a constituent of a system translates to a constraint that holds for the system.
- And using *f*-Elim we have that any constraint about the whole system gives a constraint about the components (i.e. those components that are part of a system token).

f-Intro and *f*-Elim (Example)

- Consider again the diagram



- Given the properties of the two rules we are allowed to reason along the informorphisms, since validity of constraints concerning connected tokens cannot be lost. So if flashlights with pressed switches make light, we know from the switch being pressed that the bulb is on. *f*-Intro maps the switch type to a type in a relevant system constraint the right hand side of which is mapped by *f*–Elim to a bulb type.

Local Logics (Outline)

- To make this reasoning precise we need the concept of a *local logic*, the idea is:
- A local logic $\boldsymbol{L}$=<A, $\vdash_{\boldsymbol{L}}$, $N_{\boldsymbol{L}}$> consists of a classification A, a set of sequents $\vdash_{\boldsymbol{L}}$ (satisfying some structural rules [Identity, Weakening, Global Cut]) involving the types of A, the *constraints* of $\boldsymbol{L}$, and a subset $N_{\boldsymbol{L}} \subseteq U$ (U being the domain of A), the *normal tokens* of $\boldsymbol{L}$, which satisfy $\vdash_{\boldsymbol{L}}$.

Local Logics (Adequacy)

- A local logic $\boldsymbol{L}$ is *sound* if every token is normal. It is *complete* if every sequent that holds of all normal tokens is in the consequence relation $\vdash_{\boldsymbol{L}}$. A sound and complete local logic is a classification with a *regular theory* (i.e. a theory with the mentioned structural properties).
- Reasoning at a distance involves "moving around" local logics. The inverse of a complete local logic is complete itself.

Moving Around Local Logics

- Given an informorphism between A and C and a logic ***L*** on one we obtain a natural logic on the other. If ***L*** is a logic on A, $f[\boldsymbol{L}]$ is the logic on C obtained from ***L*** by f-Intro, $f^{-1}[\boldsymbol{L}]$ is the logic on A obtained from a logic ***L*** on C by f-Elim. ("$\mathrm{Log}_C(A)$" an induced logic)
- Reasoning at a distance in our diagram is:

$$\mathrm{Log}_C(B) = f^{-1}[f[\mathrm{Log}(A)]]$$

How Information Really Flows

- Moving around logics is mediated by the channel. (Remember: infomorphism can be added up.)
- Let A be the *sum* of the constituent classifications so we have $C \rightleftarrows A$. Given a logic ***L*** on the core we use f-Elim to obtain a local logic $f^{-1}[\boldsymbol{L}]$ on A. (Containing all types of constituents.)
- Usually, ***L*** being scientific, $f^{-1}[\boldsymbol{L}]$ captures IF from a user´s perspective. The local logic is the "what" of IF, the channel is the "why"

How Information Really Flows (Example)

Let C be again our flashlight classification, B and S be the bulb and switch classifications. We build a classification B+S and an informorphism $f=f_1+f_2$, so that for a flashlight token c $f^{\vee}(c)=(f_1^{\vee}(c), f^{\vee}_2(c))$ are the bulb and switch of c. Given a type ϕ of C $f^{-1}(\phi)$ is the disjoint union of $f_1^{-1}(\phi)$ and $f_2^{-1}(\phi)$. If B supports the constraint LIT $\vdash_B$ LIVE, this will be a constraint of B+S. Then by f-Intro we have a constraint f(LIT) $\vdash_F$ f(LIVE). Now F supports ILLUM $\vdash_F$ ELEC (say emitting photons entails carrying current). We might have ILLUM=f(LIT)= f_1(LIT), and ELEC= f(PRESS)= f_2(PRESS). With f-Elim we get LIT $\vdash_{B+S}$ PRESS (i.e. we know that the switch is pressed, since the bulb is lit) – what only holds for pairs that are connected by the same flashlight, these being the *normal* tokens of the logic obtained by applying f-Elim, i.e. $\mathrm{Log}_C(B+S) = f^{-1}[\boldsymbol{L}]$ for the local logic $\boldsymbol{L}$ on C.

Centre for the Study of Logic, Language, and Information

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

Manuel Bremer, Daniel Cohnitz
Information Flow and Situation Semantics
ESSLLI 2002