Algorithmic Aspects of Analysis, Prediction, and Control in Science and Engineering: Symmetry-Based Approach

Jaime Nava

Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968
jenava@miners.utep.edu

1. Need for Analysis, Prediction, and Control in Science and Engineering

- Prediction is one of the main objectives of science and engineering.
- *Example:* in Newton's mechanics, we want to predict the positions and velocities of different objects.
- Once we predict events, a next step is to *influence* these events, i.e., to *control* the corresponding systems.
- In this step, we should select a control that leads to the best possible result.
- To be able to predict and control a system, we need to have a good *description* of this system, so that we can:
  - use this description to *analyze* the system's behavior and
  - extract the desired prediction and control algorithms from this analysis.

2. Symmetry: a Fundamental Property of the Physical World

- *One of the main objectives of science:* prediction.
- *Basis for prediction:* we observed *similar* situations in the past, and we expect similar outcomes.
- *In mathematical terms:* similarity corresponds to *symmetry*, and similarity of outcomes – to *invariance.*
- *Example:* we dropped the ball, it fall down.
- *Symmetries:* shift, rotation, etc.
- In this example, we used *geometric* symmetries, i.e., symmetries that have a direct geometric meaning.

3. Example: Discrete Geometric Symmetries

- In the above example, the corresponding symmetries form a *continuous* family.
- In some other situations, we only have a *discrete* set of geometric symmetries.
- Molecules such as benzene or cubane are invariant with respect to , e.g., rotation by 60°. molecule.

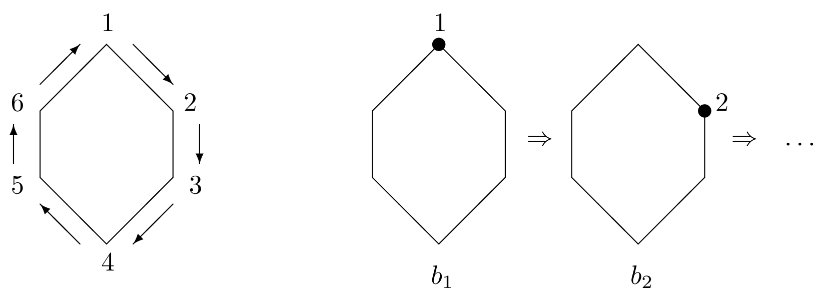

Figure 1: Benzene – rotation by 60°

4. More General Symmetries

- Symmetries can go beyond simple geometric transformations.
- *Example:* the current simplified model of an atom.
- Originally motivated by an analogy with a Solar system.
- The operation has a geometric aspect: it scales down all the distances.
- However, it goes beyond a simple geometric transformation.
- In addition to changing distances, it also changes masses, velocities, replaces masses with electric charges, etc.

5. Basic Symmetries: Scaling and Shift

- To understand real-life phenomena, we must perform appropriate measurements.
- We get a numerical value of a physical quantity, which depends on the *measuring unit.*
- *Scaling:* if we use a new unit which is λ times smaller, numerical values are multiplied by λ: $x \to \lambda \cdot x$.
- *Example:* x meters $= 100 \cdot x$ cm.
- *Another possibility:* change the starting point.
- *Shift:* if we use a new starting point which is s units before, then $x \to x + s$ (example: time).
- Together, scaling and shifts form *linear transformations* $x \to a \cdot x + b$.
- *Invariance:* physical formulas should not depend on the choice of a measuring unit or of a starting point.

6. Example of Using Symmetries: Pendulum

- *Problem:* find how period T depends on length L and on free fall acceleration g on the corresponding planet.
- Originally found using Newton's equations.
- The same dependence (modulo a constant) can be obtained only using symmetries.
- There is no fixed length, so we assume that the physics don't change if we change the unit of length.
- If we change a unit of length to a one λ times smaller, we get new numerical value $L' = \lambda \cdot L$.
- If we change a unit of time to one μ times smaller, we get a new numerical value for the period $T' = \mu \cdot T$.
- Under these transformations, the numerical value of the acceleration changes as $g \to g' = \lambda \cdot \mu^{-2} \cdot g$.

7. Pendulum Example (cont-d)

- The physics does not change by simply changing the units.
- Thus, it makes sense to require that if $T = f(L, g)$, then $T' = f(L', g')$.
- Substituting $T' = \mu \cdot T$, $L' = \lambda \cdot L$, and $g' = \lambda \cdot \mu^{-2} \cdot g$ into $T' = f(L', g')$, we get $f(\lambda \cdot L, \lambda \cdot \mu^{-2} \cdot g) = \mu \cdot f(L, g)$.
- From this formula, we can find the explicit expression for the desired function $f(L, g)$.
- Indeed, let us select λ and μ for which $\lambda \cdot L = 1$ and $\lambda \cdot \mu^{-2} \cdot g = 1$.
- Thus, we take $\lambda = L^{-1}$ and $\mu = \sqrt{\lambda \cdot g} = \sqrt{g/L}$.
- For these values λ and μ, the above formula takes the form $f(1, 1) = \mu \cdot f(L, g) = \sqrt{g/L} \cdot f(L, g)$.
- Thus, $f(L, g) = \text{const} \cdot \sqrt{L/g}$ (for the constant $f(1, 1)$).

8. What is the Advantage of Using Symmetries?

- What is *new* is that we derived it without using any specific differential equations.
- We only used the fact that these equations do not have any fixed unit of length or fixed unit of time.
- Thus, the same formula is true not only for Newton's equations, but also for *any* alternative theory.
- Physical theories need to be experimentally confirmed.
- We do not need the whole Newton's mechanics theory to derive the pend. formula – only need symmetries.
- This shows that:
  - if we have an experimental confirmation of the pendulum formula,
  - this does not necessarily mean that we have confirmed Newton's equations – just the symmetries.

9. Basic Nonlinear Symmetries

- Sometimes, a system also has *nonlinear* symmetries.
- If a system is invariant under f and g, then:
  - it is invariant under their composition $f \circ g$, and
  - it is invariant under the inverse transformation f^{-1}.
- In mathematical terms, this means that symmetries form a *group*.
- In practice, at any given moment of time, we can only store and describe finitely many parameters.
- Thus, it is reasonable to restrict ourselves to *finite-dimensional* groups.
- *Question* (N. Wiener): describe all finite-dimensional groups that contain all linear transformations.
- *Answer* (for real numbers): all elements of this group are fractionally-linear $x \to (a \cdot x + b)/(c \cdot x + d)$.

10. Independence as Another Example of Symmetry

- We encounter complex systems consisting of a large number of smaller objects (or subsystems).
- *Example:* molecules that consist of a large number of atoms.
- The more subsystems we have, the more complex the corresponding models,
  - the more difficult their algorithmic analysis because in general,
  - we need to take into account possible interactions between different subsystems.
- *Symmetry:* We know that some of these subsystems are reasonably *independent*.
- The transformations performed on one of the subsystems does not change the state of the other.

11. Discrete Symmetries

- In some cases, we have a discrete set of transformations under which the object is invariant.
- *Example:* In electromagnetism, the flas. do not change if we simply replace all pos. charges with neg. ones:
  - particles with opposite charges will continue to attract each other and
  - particles with the same charges will continue to repel each other with exactly the same force.
- Similarly, it usually does not matter whether we take, as a basis, a certain property
  - P (such as "small") or its negation
  - $P' \stackrel{\text{def}}{=} \neg P$ (such as "large").
- We can easily transform the corresponding formulas into one another.

12. Symmetries and Optimization

- It is natural to require that the model be invariant with respect to the corresponding symmetries.
- In many such cases, this invariance requirement enables us to determine the model.
- Sometimes, unlikely that the corresponding model is invariant with respect to the symmetries.
- In this case, we don't restrict to a small class of possible models.
- Out of all possible models, it is necessary to select the one which is, in some reasonable sense, the best – e.g.,
  - the most accurate in describing the real-life phenomena, or
  - the one which is the fastest to compute, etc.

13. Symmetries and Optimization (cont-d)

- What does the "best" mean?
- On the set of all appropriate models, there is a relation $\succeq$ describing which model is better or equal in quality.
- This relation must be transitive (but we can have two models of the same quality).
- We require that this relation be *final* in the sense that it should define a unique *best* model $A_{\rm opt}$, for which

$$\forall B\,(A_{\rm opt} \succeq B).$$

- If none of the models is the best, then this criterion is of no use – so there should exist optimal models.
- If *several* different models are equally best, then we can use this ambiguity to optimize something else.
- It is also reasonable to require that the relation $A \succeq B$ should be invariant relative to natural symmetries.

14. Symmetries and Optimization: Result

- Let $\mathcal{A}$ be a set, and let G be a group of transformations defined on $\mathcal{A}$.
- By an *optimality criterion*, we mean a *pre-ordering* (i.e., a transitive reflexive relation) $\preceq$ on the set $\mathcal{A}$.
- An optimality criterion is called G-*invariant* if

$$\forall g \in G\, \forall A, B \in \mathcal{A}\, (A \preceq B \Rightarrow g(A) \preceq g(B)).$$

- An optimality criterion is called *final* if there exists one and only one element $A_{\rm opt} \in \mathcal{A}$ for which

$$\forall B\, (B \preceq A_{\rm opt}).$$

- **Proposition.** *Let $\preceq$ be a G-invariant and final optimality criterion; then, $A_{\rm opt}$ is G-invariant.*

15. Symmetries and Optimization: Proof

- Let us prove that the model $A_{\rm opt}$ is indeed G-invariant, i.e., that $g(A_{\rm opt}) = A_{\rm opt}$ for every transf. $g \in G$.
- Indeed, let $g \in G$.
- From the optimality of $A_{\rm opt}$, we conclude that for every $B \in \mathcal{A}$, we have $g^{-1}(B) \preceq A_{\rm opt}$.
- From the G-invariance of the optimality criterion, we can now conclude that $B \preceq g(A_{\rm opt})$.
- This is true for all $B \in \mathcal{A}$ and therefore, the model $g(A_{\rm opt})$ is optimal.
- But since the criterion is final, there is only one optimal model; hence, $g(A_{\rm opt}) = A_{\rm opt}$.
- So, $A_{\rm opt}$ is indeed invariant.
- The proposition is proven.

16. Approximate Symmetries

- In many physical situations, we do not have *exact* symmetries, we only have *approximate* symmetries.
- *Example:* A shape of a spiral galaxy can be reasonably well described by a logarithmic spiral.
- This description is only approximate; the actual shape is slightly different.
- Actually, most symmetries are approximate.
- In some cases, the approximation is so good that we can consider the object to be fully symmetric.
- In other cases, we need to take asymmetry into account to get an accurate description of the corr. phenomena.

17. From Methods to Results

- *Main algorithmic problems (reminder):* analysis, prediction, and control of real-life systems.
- Up to now, we described our *methodology*: the use of symmetries.
- Let us now show the *results* of using symmetries in all algorithmic problems of sciences and engineering.
- First, we show how the symmetry-based approach can be used in the *analysis* of real-life systems.
- Then, we show how this approach can be used in the algorithmics of *prediction*.
- Finally, we show how the symmetry-based approach can be used in the algorithmics of *control*.

18. Algorithmic Aspects of Real-Life Systems Analysis: Symmetry-Based Approach

- Our main objectives are to *predict* the systems' behavior and to find the best way to *change* this behavior.
- In these tasks, we first need to *describe* the systems in precise terms.
- In Chapter 2, we show that *symmetries* can help in this description.
- Real-life systems consist of interacting *subsystems*.
- So, to describe a system, we must first describe *fundamental* systems: molecules, elementary particles, etc.
- In Chapter 2, we focus on the description of such fundamental systems.

19. Algorithmic Aspects of Real-Life Systems Analysis: Overview

- We start with the most natural symmetries – continuous families of geom. symmetries (rotations, shifts).
- A shape of the molecule is formed by its atoms.
- We look for symmetries, i.e., for transformations that preserve the shape of a molecule.
- H_2: invariant w.r.t. *all* rotations around its axis; we have a *continuous* family of angles from 0 to 360°.
- Benzene: only inv. w.r.t. *discrete* angles 60°, 120°, ...
- Except for linear molecules like H_2, a molecule cannot have a continuous family of symmetries.
- For a molecular shape to allow a continuous family of symmetries, it must contain a large number of atoms.
- Such molecules are typical in biosciences.

20. Algorithmic Aspects of Real-Life Systems Analysis: Overview (cont-d)

- In Section 2.2, we show how, for biomolecules, the corr. symmetries naturally explain the observed shapes.
- Smaller molecules, as we have mentioned, can only have discrete symmetries.
- In Section 2.3, we show how the symmetries approach can help in describing such molecules.
- Finally, on the quantum level of elementary particles, we, in general, do not have geometric symmetries.
- Instead, we have a reasonable symmetry-related physical idea of *independence*.
- We show that this idea leads to a formal justification of quantum theory (in Feynman integral form).
- In all these cases, symmetries help.

21. Algorithmic Aspects of Real-Life Systems Analysis: Details

- *Application:* explaining shapes of secondary elements in protein structure.
- Protein structure is invariably connected to protein function.
- There are two important secondary structure elements: alpha-helices and beta-sheets.
- Their actual shapes can be complicated, but usually approx. by cylindrical spirals, planes and cylinders.
- *Result:* these geometric shapes are indeed the best approximating families for secondary structures.
- This result expands on the ideas pioneered by a renowned mathematician M. Gromov.

22. Algorithmic Aspects of Real-Life Systems Analysis: Details (cont-d)

- *Application:* understanding properties of molecules with variant ligands.
- Molecules can be obtained from a "template" molecule by replacing some of its atoms with *ligands*.
- Testing of all possible replacements is time-consuming.
- Avoid by testing some of the replacements and then extrapolate to others.
- D. J. Klein and co-authors proposed to use a poset extrapolation technique developed by G.-C. Rota.
- *Limitation:* technique originally proposed on a heuristic basis.
- No convincing justification of its applicability to chemical (or other) problems.

23. Algorithmic Aspects of Real-Life Systems Analysis: Details (cont-d)

- Previously, we showed that the poset technique is equivalent to Taylor series extrapolation.
- A more familiar (and much more justified) technique.
- *Results:*
  - The equivalence with Taylor series extr. can be extended to the case of variant ligands.
  - This approach is also equivalent the Dempster-Shafer approach.

24. Algorithmic Aspects of Prediction: Symmetry-Based Approach

- As we have mentioned earlier, one of the main objectives of science is to predict future events.
- From this viewpoint, the first question that we need to ask is: *is it possible* to predict?
- In many cases, predictions are possible.
- In many other practical situations, what we observe is a *random* (un-predictable) sequence.
- The question of how to check whether a sequence is random is analyzed in Section 3.2.
- In this analysis, we use symmetries – namely, we use scaling symmetries; see below.
- In situations when prediction is, in principle, possible, the next questions is: *how* can we predict?

25. Algorithmic Aspects of Prediction (cont-d)

- In cases when we know the corresponding equations, we can use these equations for prediction.
- In many practical situations, however, we do not know the equations.
- In such situations, we need to use general prediction and extrapolation tools, e.g., neural networks.
- In Section 3.3, we show how discrete symmetries can help improve the efficiency of neural networks.
- Once the prediction is made, the next question is *how accurate* is this prediction?
- In Section 3.4, we show how scaling symmetries can help in *quantifying* the uncertainty of the corr. model.
- In Section 3.5, we use similar symmetries to come up with an optimal way of *processing* uncertainty.

26. Algorithmic Aspects of Prediction (cont-d)

- In Sect. 3.6, on a geophysical example, we estimate the accuracy of *spatially locating* the measurement results.
- In practice, we often need to have the prediction results by a certain time.
- It is then important to perform the corresponding computations efficiently – by the deadline.
- The theoretical possibility of such efficient computations is analyzed in Section 3.7.
- Overall, we show that symmetries can help with all the algorithmic aspects of prediction.

27. Example: How to Check Randomness

- A non-random sequence like 010101... can be generated by a short program.
- If a sequence is truly random, the only way to generate it is to print it bit by bit.
- The shortest length of a program that computes s is called its *Kolmogorov complexity* $K(s)$.
- Thus, a sequence is random if and only if its Kolmogorov complexity is close to its length.
- The big problem is that the Kolmogorov complexity is, in general, *not* algorithmically *computable*.
- Thus, it is desirable to come up with *computable* approximations.

28. Approximating Kolmogorov Complexity

- *Reminder:* we need to approximate Kolmogorov complexity $K(s)$.
- At present, most algorithms for approximating $K(s)$:
  - use some loss-less compression technique to compress s, and
  - take the length $\widetilde{K}(s)$ of the compression as the desired approximation.
- However, this approximation has limitations: for example,
  - in contrast to $K(s)$, where a change (one-bit) change in s cannot change $K(s)$ much,
  - a small change in s can lead to a drastic change in $\widetilde{K}(s)$.

29. Other Applications of Kolmogorov Complexity

- Kolmogorov complexity is useful beyond checking randomness.
- E.g.: we can check how close are two DNA sequences s and s' by comparing $K(ss')$ with $K(s)+K(s')$:
  - if they are *unrelated*, the only way to generate ss' is to generate s and then generate s', so:
  $$K(ss') \approx K(s)+K(s');$$
  - if they are *related*, we have $K(ss') \ll K(s)+K(s')$.
- By taking strings s and s' describing the same text in languages L, L', we can check how close are L and L'.

30. I-Complexity

- Limitation of $\widetilde{K}(s)$: a small change in $s = (s_1 s_2 \ldots s_n)$ can lead to a drastic change in $\widetilde{K}(s)$.
- To overcome this limitation, V. Becher and P. A. Heiber proposed the following new notion of *I-complexity*.
- For each i, we find the largest $B_s[i]$ of the length ℓ of strings $s_{i-\ell+1} \ldots s_i$ which are substrings of $s_1 \ldots s_{i-1}$.
- For example, for $aaaab$, the corresponding values of $B_s(i)$ are 01230.
- We then define $I(s) \stackrel{\rm def}{=} \sum\limits_{i=1}^{n} f(B_s[i])$, for an appropriate decreasing function $f(x)$.
- Specifically, it turned out that the *discrete derivative of the logarithm* works well: $f(x) = \text{dlog}(x+1)$, where

$$\text{dlog}(x) \stackrel{\rm def}{=} \log(x+1) - \log(x).$$

31. Good Properties of I-Complexity

- *Reminder:* $I(s)=\sum_{i=1}^{n} f(B_s[i])$, where:
  - $B_s[i]$ is the the largest length ℓ of strings $s_{i-\ell+1}\ldots s_i$ which are substrings of $s_1\ldots s_{i-1}$, and
  - $f(x)=\log(x+1)-\log(x)$.
- *Similarly to $K(s)$:*
  - If s starts s', then $I(s)\le I(s')$.
  - We have $I(0s)\approx I(s)$ and $I(1s)\approx I(s)$.
  - We have $I(ss')\le I(s)+I(s')$.
  - Most strings have high I-complexity.
- *In contrast to $K(s)$:* I-complexity can be computed in linear time.
- *A natural question*: why this function $f(x)$?

32. Towards Precise Formulation of the Problem

- We take $f(x) = F(x+1) - F(x)$, for some $F(x)$.
- V. Becher and P. A. Hieber selected $F(x) = \log(x)$.
- There are not much symmetries on integer values x, but we can extend $F(x)$ to all real numbers.
- Which monotonic function $F(x)$ from reals to reals should we choose?
- *Reminder:* in the continuous case, the numerical value of each quantity depends:
  - on the choice of the measuring unit and
  - on the choice of the starting point.
- By changing them, we get a new value $x' = a \cdot x + b$.
- For length x, the starting point 0 is fixed.
- So, we only have re-scaling $x \to x' = a \cdot x$ (e.g., bits vs. bytes).

33. Our Result

- By changing a measuring unit, we get $x' = a \cdot x$.
- When we thus re-scale x, the value $y = F(x)$ changes, to $y' = F(a \cdot x)$.
- It is reasonable to require that the value y' represent the same quantity.
- So, we require that y' differs from y by a similar re-scaling:

  $y' = F(a \cdot x) = A(a) \cdot F(x) + B(a)$ for some $A(a)$ and $B(a)$.

- It turns out that all monotonic solutions of this equation are linearly equivalent to $\log(x)$ or to x^α, i.e.:

$$F(x) = \widetilde{a} \cdot \ln(x) + \widetilde{b} \text{ or } F(x) = \widetilde{a} \cdot x^\alpha + \widetilde{b}.$$

- So, symmetries do explain the selection of the function $F(x)$ for I-complexity.

34. Proof

- *Reminder:* for some monotonic function $F(x)$, for every a, there exist values $A(a)$ and $B(a)$ for which

$$F(a \cdot x) = A(a) \cdot F(x) + B(a).$$

- *Known fact:* every monotonic function is almost everywhere differentiable.
- Let $x_0 > 0$ be a point where the function $F(x)$ is differentiable.
- Then, for every x, by taking $a = x/x_0$, we conclude that $F(x)$ is differentiable at this point x as well.
- For any $x_1 \neq x_2$, we have $F(a \cdot x_1) = A(a) \cdot F(x_1) + B(a)$ and $F(a \cdot x_2) = A(a) \cdot F(x_2) + B(a)$.
- We get a system of two linear equations with two unknowns $A(a)$ and $B(a)$.

35. Proof (cont-d)

- We get a system of two linear equations with two unknowns $A(a)$ and $B(a)$:

$$F(a\cdot x_1)=A(a)\cdot F(x_1)+B(a).$$

$$F(a\cdot x_2)=A(a)\cdot F(x_2)+B(a).$$

- Thus, both $A(a)$ and $B(a)$ are linear combinations of differentiable functions $F(a\cdot x_1)$ and $F(a\cdot x_2)$.
- Hence, both functions $A(a)$ and $B(a)$ are differentiable.
- So, $F(a\cdot x)=A(a)\cdot F(x)+B(a)$ for differentiable functions $F(x)$, $A(a)$, and $B(a)$.
- Differentiating both sides by a, we get

$$x\cdot F'(a\cdot x)=A'(a)\cdot F(x)+B'(a).$$

- In particular, for $a=1$, we get $x\cdot\dfrac{dF}{dx}=A\cdot F+B$, where $A\stackrel{\text{def}}{=}A'(1)$ and $B\stackrel{\text{def}}{=}B'(1)$.

36. Proof (final part)

- *Reminder:* $x \cdot \dfrac{dF}{dx} = A \cdot F + B.$
- So, $\dfrac{dF}{A \cdot F + b} = \dfrac{dx}{x}$; now, we can integrate both sides.
- *When* $A = 0$*:* we get $\dfrac{F(x)}{b} = \ln(x) + C$, so
$$F(x) = b \cdot \ln(x) + b \cdot C.$$
- *When* $A \neq 0$*:* for $\widetilde{F} \stackrel{\text{def}}{=} F + \dfrac{b}{A}$, we get $\dfrac{d\widetilde{F}}{A \cdot \widetilde{F}} = \dfrac{dx}{x}$, so $\dfrac{1}{A} \cdot \ln(\widetilde{F}(x)) = \ln(x) + C$, and $\ln(\widetilde{F}(x)) = A \cdot \ln(x) + A \cdot C$.
- Thus, $\widetilde{F}(x) = C_1 \cdot x^A$, where $C_1 \stackrel{\text{def}}{=} \exp(A \cdot C)$.
- Hence, $F(x) = \widetilde{F}(x) - \dfrac{b}{A} = C_1 \cdot x^A - \dfrac{b}{A}.$
- The theorem is proven.

37. Algorithmic Aspects of Control: Symmetry-Based Approach

- Up to now, we concentrated on the problem of *predicting* the future events.
- Once we are able to predict future events, a natural next step is to *control* the corresponding system.
- In this step, we should select a control that leads to the *best* possible result.
- Sometimes, we know the exact equations, so we can simply use known optimization techniques.
- Often, exact equations are not known, so we have the use the knowledge of human experts.
- Many such intelligent techniques are *empirical*, their results which are far from optimal.

38. Algorithmic Aspects of Control (cont-d)

- In Chapter 4, we show that symmetry-based techniques can be very useful in improving these results.
- We will illustrate this usefulness on all level of the problem of selecting the best control.
- First, on a strategic level, we need to select the best class of strategies.
- In Sect. 4.2, we derive the best class of strategies for *fuzzy control*, an important class of intelligent controls.
- In the corr. derivation, we use logical symmetries – the symmetry between true and false values.
- Once a class of strategies is selected, we need to select the best strategy within a given class.

39. Algorithmic Aspects of Control (cont-d)

- Once a class of strategies is selected, we need to select the best strategy within a given class:
  - in Section 4.3, we use approximate symmetries to find the best implementation of fuzzy control;
  - in Section 4.4, we show that the optimal selection of operations leads to a symmetry-based solution.
- Often, we have several strategies coming from different aspects of the problem.
- We need to combine these strategies into a single strategy that takes all the aspects into account.
- In Section 4.5, we use logical symmetries to find the best way of combining the resulting fuzzy decisions.
- Overall, we show that symmetries can help with all the algorithmic aspects of control.

40. Example: Selecting the Best Exclusive-Or Operations

- Intelligent control comes from expert statements.
- Expert statements often use logical connectives like "or".
- In natural language, "or" sometimes means "inclusive or" and sometimes means "exclusive or".
- Traditionally, intelligent control uses "inclusive or" (t-conorms).
- We want to adequately describe commonsense and expert knowledge.
- Therefore it is important to also have fuzzy "exclusive or" operations $f_\oplus(a, b)$.

41. Selecting Exclusive-Or Operations (cont-d)

- *Reminder:* it is important to have fuzzy "exclusive or" operations $f_\oplus(a,b)$.
- *Main idea:* The degrees of certainty are only approximately defined.
- It is reasonable to require that the operation be the *least sensitive* to small changes in the inputs.
- This requirement corresponds to *approximate symmetry*.
- *Result:* the least sensitive fuzzy "exclusive or" operation has the form

$$f_\oplus(a,b) = \min(\max(a,b), \max(1-a, 1-b)).$$

42. Possible Ideas for Future Work

- In this dissertation, on numerous examples, we showed that symmetries help in science and engineering.
- The breadth and depth of these examples show that symmetry-based approach is indeed very promising.
- However, to make this approach more widely used, additional work is needed.
- Indeed, in each of our examples, the main challenge is finding the relevant symmetries.
- As of now, we have found these symmetries on a *case-by-case* basis.
- It would be great to develop a *general* methodology of finding the relevant symmetries.
- Such a general methodology would help to apply symmetry-based approach to important new practical problems.

43. Acknowledgements

- I would like to thank Dr. Vladik Kreinovich for his teachings and for pushing me with professionalism.
- I also wish to thank Dr. Luc Longpré and Dr. Larry Ellzey for their supervision, input, and expert advice.
- Thanks to Drs. Benoit Bagot, Art Duval, Yuri Gurevich, Patricia Nava, and James Salvador for their help.