Simplification and Abstraction of Kinematic Behaviors

Leo Joskowicz
IBM T. J. Wat son Research Center
P.O. Box 704
Yorktown Heights, NY 10598
E-mail: josko@IBM.COM

Abstract

Efficient problem-solving in any physical domain requires: (1) the ability to ignore irrelevant information by incorporating constraints and assumptions (simplification), and (2) the ability to ignore detail by reasoning at different levels of resolution (abstraction). Existing analysis methods for mechanical devices derive qualitative descriptions of a mechanism's kinematic behavior from the shapes and positions of its parts. Although qualitative, these descriptions provide a single level of abstraction which is exceedingly complex and detailed to automate common analysis and design tasks. This paper presents a set of operators to simplify and abstract kinematic descriptions derived from configuration spaces. The description hierarchy defined by these operators provides a basis for many reasoning tasks, such as mechanism comparison - determining when two mechanisms are kinematically equivalent.

1 Introduction

Recent methods for the analysis of mechanical devices derive descriptions of a mechanism's kinematic behavior from the shape and the initial position of its parts [Faltings, 87a; 87b], [Nielsen, 88a; 88b], [Joskowicz, 87, 88; 89a]. Although qualitative, these descriptions provide a single level of abstraction which is often too detailed and complex to be of practical use in automated design. The complexity of the behavioral descriptions arises from two factors: (1) from the (local) kinematic pair descriptions, as a result of the objects' shape complexity, and (2) from the overall (global) mechanism descriptions, as a result of the combinatorial complexity of possible object positions.

Local descriptions are derived from two-dimensional configuration spaces[1] defined by the objects' degrees of freedom. The configuration space of a pair of objects is computed by analyzing all pairwise contacts between object features (the vertices and edges of the objects' contours). Each pairwise contact defines a half-space bounded by a one-dimensional curve (called a *configuration space boundary* or CS-boundary for short) that separates free and forbidden placements. The intersection of these half spaces defines the components of the configuration space, which are then partitioned into regions. The regions, defined by monotone CS-boundary segments, reflect the qualitatively different behaviors of the pair. There can be as many regions as there are possible contacts; as a result, behavioral description are frequently too detailed.

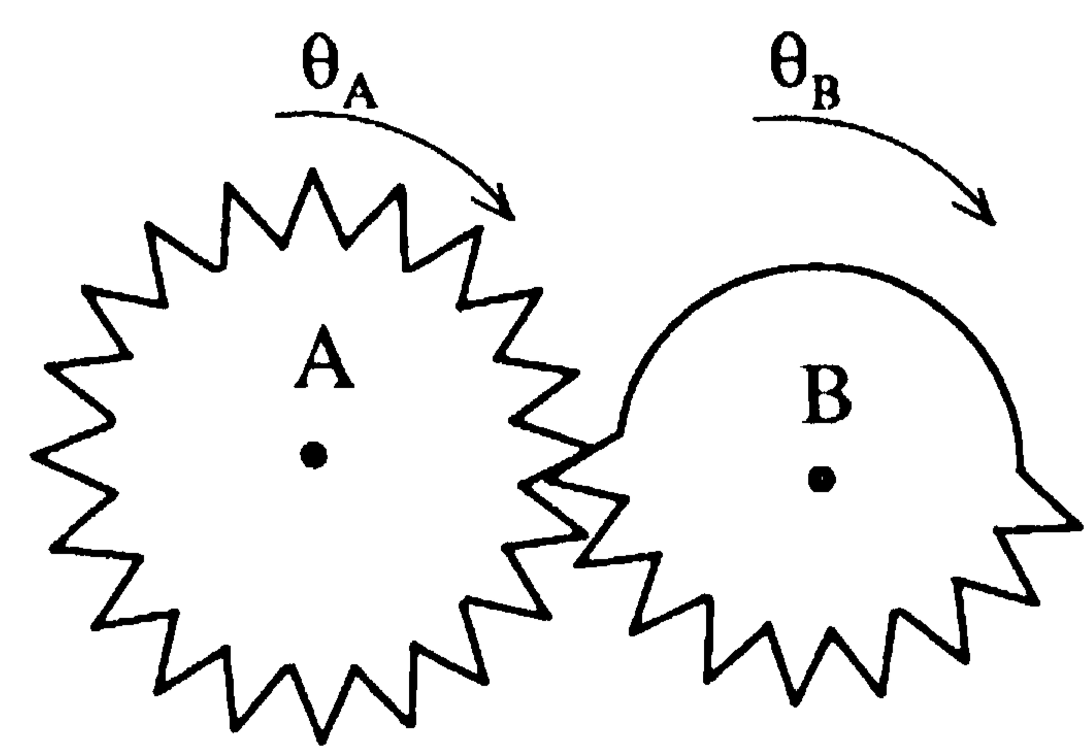

Figure 1: The Half-Gear Pair

Consider the half-gear pair in Fig. 1, consisting of a 20-teeth gear and a 9-teeth half gear of equal diameter pinned at their centers to a fixed frame. Their configuration space is shown in Fig. 2(a): dark areas correspond to forbidden object positions; solid lines correspond to CS-boundaries. Fig. 2(b) shows a detailed view of the CS-boundaries and the partition into regions[2].

Region *RQ* corresponds to positions in which the two gears are disengaged, and thus their rotations are independent; regions r, correspond to positions in which the two gears are meshed. Each r, region is delimited by two CS-boundaries, corresponding to the upper and lower contact between two teeth. When a contact occurs, the rotating one gear causes the rotation of the other in the opposite direction; the ratio between both rotations is given by the function defining the CS-boundary. The fragmented nature of the CS-boundary reflects the teeth

*The configuration space of a mechanism defines the set of *free placements* (position and orientations) of objects so that no two objects overlap [Lozano-Perez, 83].

[2]Both figures adapted from [Faltings, 87a]; for clarity, only half of the "strips" are shown in (a).

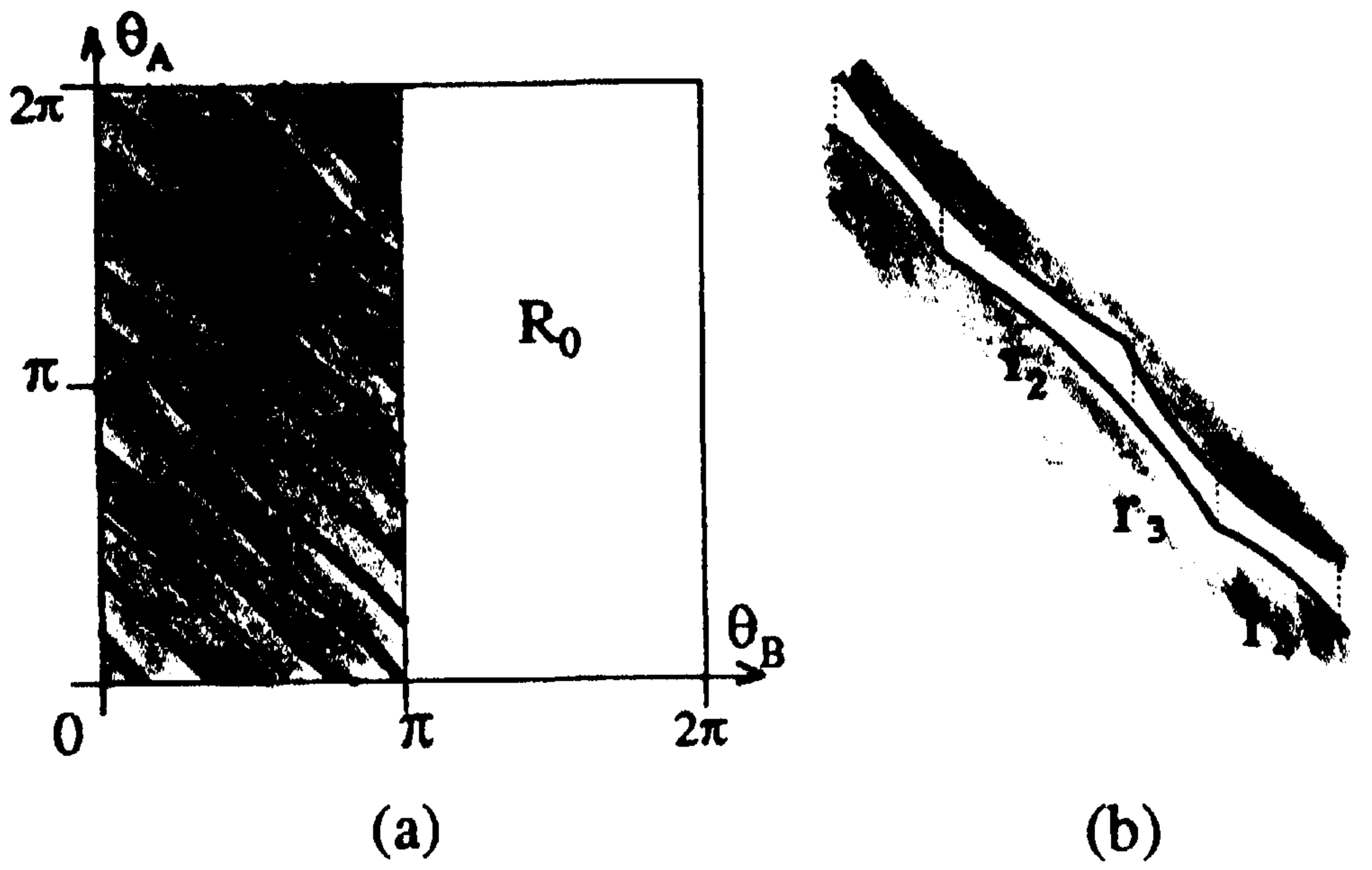

Figure 2: The Configuration Space of the Half Gear Pair

contact changes, and indicates slight variations in the motion transmission ratio. The small intervals between CS-boundaries in each region corresponds to gear interplay (backlash), Since there are two regions for each pairwise tooth contact, there are 20 x 9 x 2 = 360 r, regions; the regions are qualitatively different due to angular offset and interval differences. This kinematic description is clearly too detailed for a CAD system designing a gearbox with many gear pairs.

Global behaviors are obtained by composing all local descriptions. In general, composing local descriptions requires algebraic techniques. However, for fixed axes mechanisms, in which objects can only move along axes that are fixed in space, the composition requires a small set of symbolic rules [Joskowicz, 87]. In the worst case, the composition results in the cross-product of all local regions for every pair, producing an overall mechanism description with exponentially many regions.

Consider the cylinder lock in Fig. 3, consisting of a cylinder *C* rotationally mounted on a fixed frame *F*. Five pairs of pins of different lengths, (P_t, Q_i), are mounted inside five aligned cylindrical holes in the cylinder and the frame. The pins are kept in contact by springs and can only translate along the axes of the holes; their role is to prevent the rotation of the cylinder. When the right key is inserted, the pins are raised so that the top of the lower pins and the bottom of the upper pins coincide exactly with the outer surface of the cylinder (Fig. 3(b)). When the pins are aligned, the cylinder rotates together with the key, as there are no obstacles to prevent this rotation. The cylinder then pushes a tumbler (not shown in the figure) that locks the door by preventing it from rotating around its axis.

The analysis identifies three qualitatively different positions of the pins: (1) the upper and lower pins are in contact with the cylinder, preventing it from rotating; (2) the upper and lower pins are in contact with the fixed frame, also preventing the cylinder from rotating; (3) the top of the lower pin, the bottom of the upper pin, and the top of the cylinder are aligned; the cylinder is then free to rotate, in which case the pins cannot translate. The key/cylinder pair has two characteristic

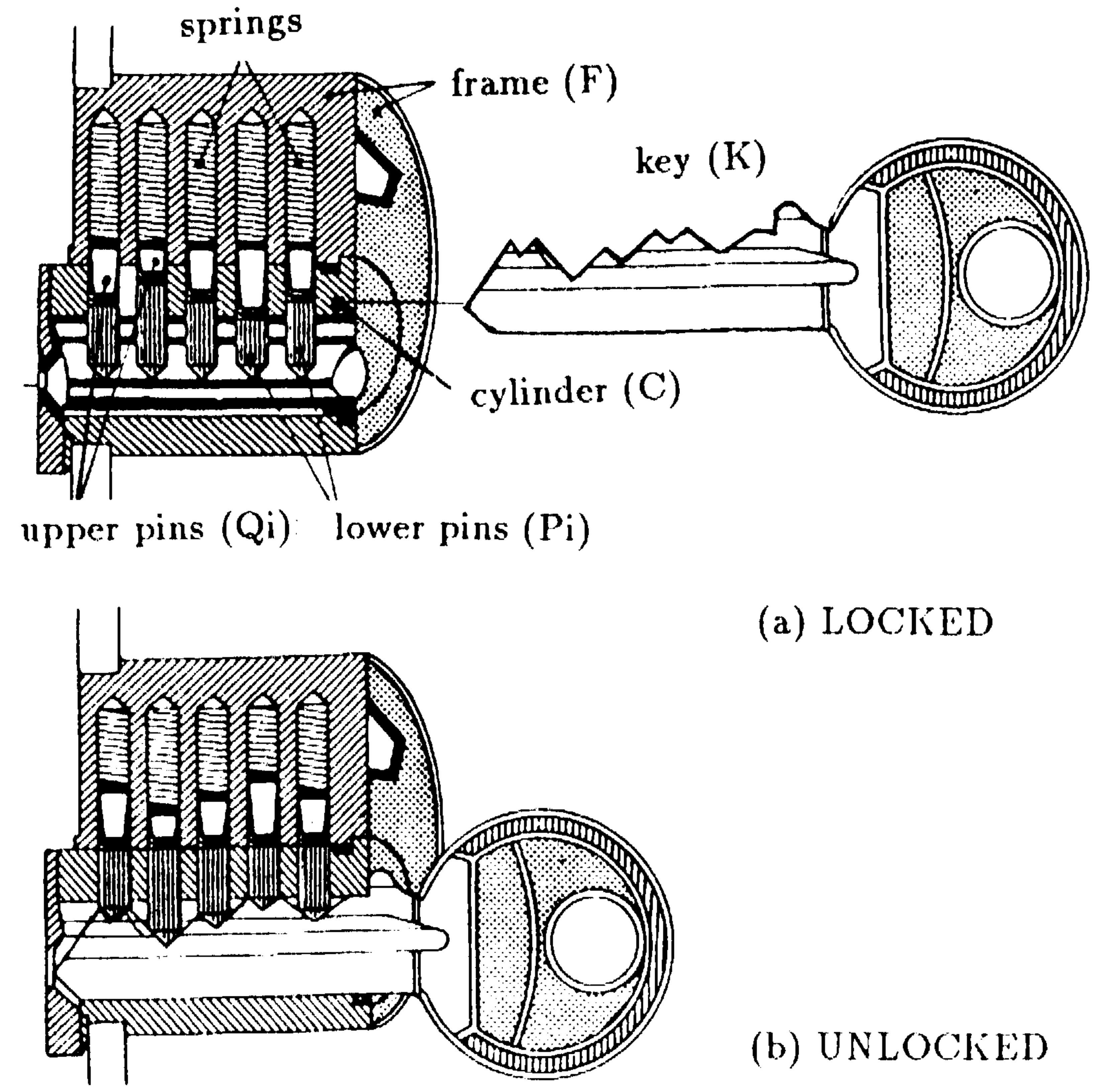

Figure 3: The Cylinder Lock

positions: (1) the key and the cylinder are not in contact; the rotation of the cylinder is independent from the rotation and/or translation of the key; (2) the key is inside the cylinder; the key is free to translate inside the cylinder, but the rotations of the key and cylinder are dependent. The global description contains many unreachable regions and irrelevant detail. For example, 3^5 = 243 regions correspond to all the possible combinations of pin positions when the key is outside the cylinder. But these positions can only be reached when the pins are directly pushed from the outside! Assuming that input motions are only applied to the key, all these regions are unreachable. Another source of detail comes from the regions created by the pin positions as the key is inserted: the pins follow the upper contour of the key, thereby changing characteristic positions. All these distinctions are irrelevant if we are only interested in the behaviors of the key and the cylinder.

Two ideas play a key role in coping with the complexity and focus of behavioral descriptions: *simplification* and *abstraction.* Simplification incorporates additional information in the form of assumptions and constraints. Abstraction ignores irrelevant detail by defining different levels of resolution. This paper defines a set of simplification and abstraction *operators* for kinematic descriptions.

2 Describing Kinematic Behavior

This section presents the symbolic language used to describe kinematic behaviors. Objects in a fixed axes mechanism can only rotate or translate (or both) along axes that are fixed in space. We can thus classify all their possible motions along a fixed axis into five categories: no motion, rotation, translation, independent rotation and translation, and coordinated rotation and translation. To each category, we associate a predicate indicating the type of motion and the axis along which the motion takes place. To describe the extent of the motion, we associate *motion parameters* bound by intervals defining the motion's range. The type of motion, together with the motion parameters and their intervals form a *possible motion label* which completely describes an object's kinematic behavior. The five possible motion labels for object *A* along axis *O* are: *fixed*(A,O,p), *p .rot*(A,O,θ), *jranslation*(A,O,X), *P-cyUnder* (A,O,X,θ), and *pJielical*(A,O,X,θ).

Relationships between object motions, indicating how objects constrain each other's motion through contact, are specified by motion parameter relations. Mechanism behaviors are described by a set of possible motion labels (one for each object) and the dependencies between their motion parameters. Such a description is called a *possible motions region* of a mechanism's behavior[3]. A *region diagram* describes all the possible kinematic behaviors of a mechanism and the transitions between them. The region diagram is an undirected graph whose nodes represent possible motion regions and whose edges represent possible transitions between regions.

To distinguish between qualitatively different behaviors, all regions in the diagram must be qualitatively different. Two possible motions regions, *Ri* and *Rj,* are qualitatively different iff at least one of the following holds: (1) the motion type of at least one object is different in *Ri* and *Rj;* (2) the motion parameter intervals defining *Ri* and *Rj* cannot be merged into continuous intervals forming a new region *Rk* = RU *Rj*; (3) motion parameter relations in *Ri* and *Rj* are not identical; (4) motion parameter relations in *Ri* and *Rj* are identical but at least one of these relations is monotonically increasing in one region and monotonically decreasing in the other. Region diagrams constitute a symbolic, qualitative description of all the possible kinematic behaviors of a mechanism.

3 Simplification and Abstraction Operators

We define simplifications and abstractions as operations on kinematic descriptions and their underlying configuration spaces. To be meaningful, these operations should not introduce spurious behaviors or alter the original relations between behaviors; in other words they should be sound and complete. Soundness guarantees that no new behaviors are introduced, whereas completeness guarantees that no possible behaviors were lost. A third property, compositionality, guarantees that the simplification/abstraction can be applied to the local pairwise descriptions *before* composing them. The following two subsections briefly describe the operators to simplify and abstract kinematic behaviors. For detailed algorithms and complexity analyses, see [Joskowicz, 89b].

[3]These descriptions are called regions because they correspond to *regions* of the mechanism's configuration space.

3.1 Local Operators

Local operators simplify and abstract kinematic pair descriptions; they are applied to their two-dimensional configuration spaces defined by n monotone CS-boundary segments. Local operators are applied immediately after the configuration space is computed. There are ten local operators, all sound and complete:

Kinematic Constraints Simplifications: specify the object that causes the motion, the type of motion, and the extent of the motion (Input-Part, Input-Type, and Input-Range, respectively). These additional constraints, derived from the context in which the pair is operating, rule out possible behaviors that become unreachable as a consequence of this constraining information. Input-Part transforms every region *R* into a new set of regions, r;. In the new regions, the motion parameter of the passive object changes only when the active object's motion parameter changes; there is one ri for every region adjacent to *R.* Input-Type reduces the number of pairwise configuration spaces to be analyzed (one for every pair of fixed axes). Input-Range rules out behaviors that are outside the specified motion range by deleting regions outside the range, and restricting those who lie on the boundary. All three simplifications can be implemented in *O(n).*

Dynamic Constraints Simplifications: account for the action of gravity, springs, and friction, thereby restricting the possible motions of objects. Three kinematic constraints are used to model these dynamic constraints: (1) a constant contact relation between two objects; (2) a preferred (or default) position of an object when it is not subject to other contact constraints, and; (3) conditions on the motion relation parameters restricting motion transmission. The first two constraints model the effects of springs and gravity. A force applied to an object causes it to remain in contact with its neighboring objects in the direction of the force. Thus, all positions in which the two objects are not in contact can be ruled out. When no contact occurs, the object will move in the direction of the force until it reaches a stable position. To avoid introducing time information, we assume that motions due to forces are infinitely faster than input motions. The third constraint models friction by restricting motion transmission through contacts whose CS-boundary tangent at the point of contact is smaller than a predefined coefficient //, regardless of the force's magnitude and objects' masses. These constraints rule out possible transitions between regions and reduces region intervals. All three simplifications can be implemented in *0(n).*

Linearization Abstraction: approximates the exact motion parameter relations with pieccwise linear relations. Every CS-boundary is divided into monotonically continuous segments, which are then replaced by a set of lines. The new linear CS-boundary might intersect other CS-boundaries, thereby producing a topologically inconsistent abstraction. This is avoided by further subdividing the CS-boundary into smaller segments. Assuming that each segment is broken only a constant number of times, linearization can be implemented using an efficient line intersection algorithm in *Q(n*logn).

Qualitative Abstraction: groups contiguous behaviors with similar monotone motion parameter relations into one behavior. This abstraction hides the contact details that produce fragmented behavioral descriptions. The operator merges contiguous monotone CS-boundaries and creates *qualitative CS-boundaries.* These boundaries indicate the qualitative motion parameter relation (monotonically increasing or decreasing) in the new regions. Merging CS-boundaries also requires testing for intersections, and thus can be implemented in *0(n* log n). Linearization and Qualitative abstraction have been used for the design of new object shapes in kinematic pairs [Joskowicz and Addanki, 88].

Gap-Closing Abstraction: transforms behaviors with small backlash into behaviors without backlash, where object motions are tightly coupled. This operator merges CS-boundaries defining narrow "channels" of free placements reflecting backlash and tolerancing errors (Fig. 2(b)). The small positional variations inside the channels reflect negligible variations that can usually be ignored. The operator examines every region for a possible boundary merge, and can be implemented in *0(n).*

Behavior Parametrizations: these operators are not simplification or abstraction operators per se: they are designed to compare two kinematic behaviors and find a common description by parameterizing their possible motion labels. This operation is useful for detecting behavioral similarities and periodicity. Possible motions labels have three potential candidates for parameterization: region intervals, axes of motions, and motion parameter relations. Two regions can be parameterized in their intervals when: (1) their motion type is identical, and is defined along the same axis; (2) the relations between their motion parameters are identical, and (3) their interval regions are different but proportionally scaled. Similarly, two regions can be parameterized in their axes when the axes of motion are not required to be identical. Finally, two regions can be parameterized in their motion relations when these relations are linearly similar $(y \leq f(x)$ and $y \leq f(x) + c)$, scalarly similar $(y \leq f(x)$ and $y \leq c.f(x))$, have a phase difference $(y \leq f(x)$ and $y \leq f(x + c))$, or any combination of them. This operation takes constant time when applied after Linearization.

Periodicity Abstraction: detects patterns of repetitive behavior to create a common behavioral description. This description is produced by identifying parametrically isomorphic subgraphs in the region diagram. Two subgraphs are parametrically isomorphic iff they are topologically equivalent and there is a one-to-one parametric matching between regions. Although graph isomorphism can be tested efficiently - region diagrams are graphs with a planar embedding - subgraph isomorphism, even for planar graphs, is in NP-Complete, and is thus computationally expensive.

The order in which the operators are applied obeys two motivations: (1) discard as soon as possible the behaviors that, due to constraints, are unreachable, and (2) reduce the number of regions in the diagram as quickly as possible. Therefore, simplification operators are applied first, because they incorporate additional constraints.

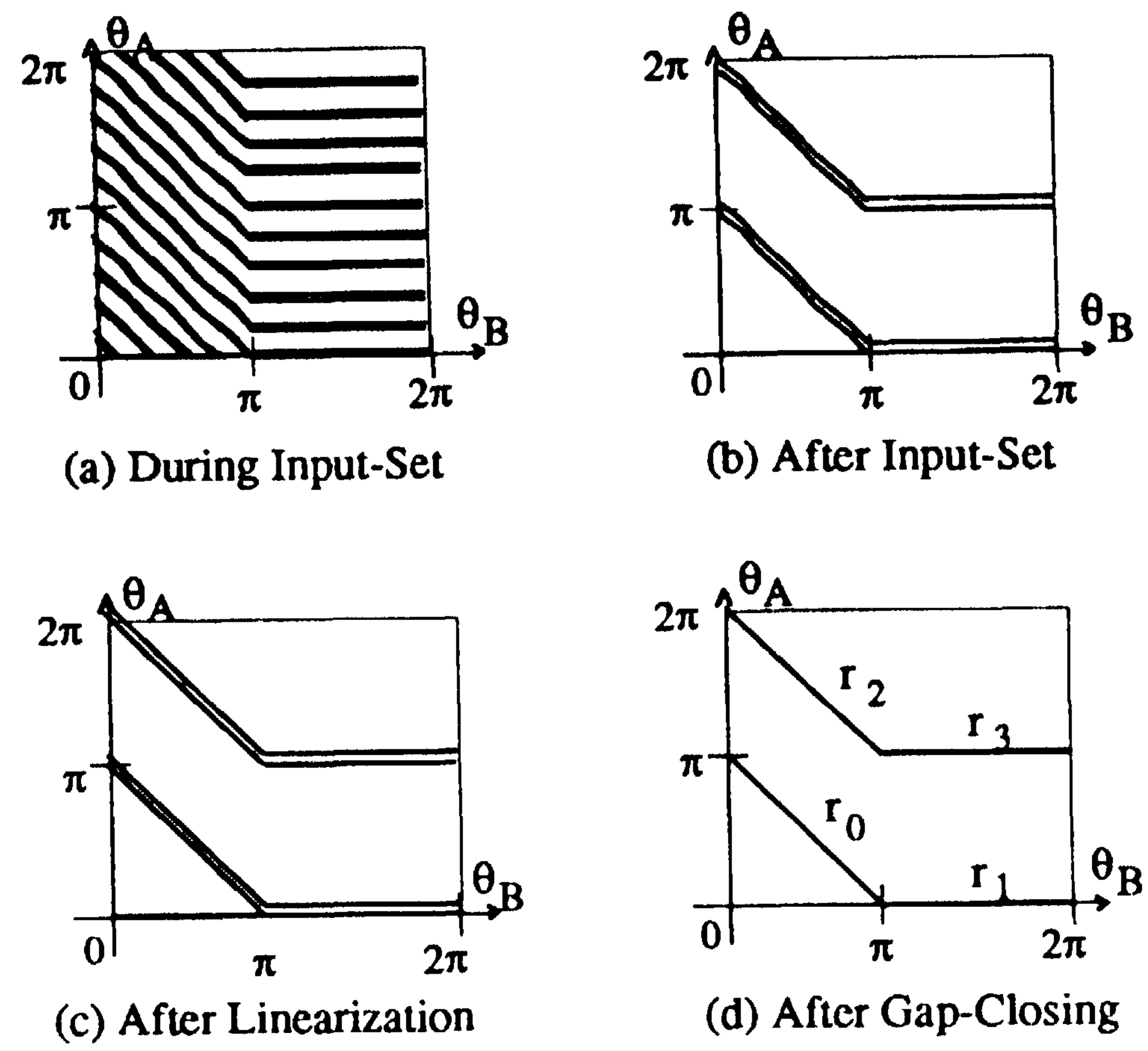

Figure 4: Configuration Space Transformations

Then, abstraction operators are applied in increasing order of coarseness: first Linearization, then Qualitative abstraction. Gap-Closing depends on the results of the previous two abstractions and is applied repeatedly until no further change occurs. Periodicity abstraction is applied at the very end, when the local region diagram is smallest.

As an example, consider the effects of local operators on the half-gear pair's configuration space of Fig. 2. We assume that input motions are only applied to gear *B* and that the initial placement is $\theta_A = \theta_B = 0$. First, Input-Part simplification reduces *RQ* to a set of disjoint two-dimensional strips bounded by the linear relations $\theta_A \geq \theta_i$ and $\theta_A \leq \theta_i + \epsilon$ in the interval $\theta_B \in [\pi, 2\pi]$. Only two strips, $\theta_i = 0$ and π, are reachable from the initial placement. As a consequence, only the regions connected to these two strips are reachable in the interval $\theta_B \in [0, \pi]$; all other regions are unreachable and are thus discarded. Linearization and Qualitative abstraction merge the remaining CS-boundaries in the interval $\theta_B \in [0, \pi]$ into two regions, each defined by two parallel lines. Gap-Closing merges the parallel lines in these regions as well as the two strips in the interval $\theta_B \in (\pi, 2\pi]$. The successive effects of these operators on the configuration space are shown in Fig. 4. Periodicity abstraction identifies two parametrically isomorphic subgraphs, (r_o,r_i) and (r2,r3), commonly described with a parameter i, where *i* = 0 describes *(r_0,r_i)* and *i* = 1 describes (r_2,r_3):

$$MESH\text{: } p\_rotation(A, O_A, \theta_A),\ p\_rotation(B, O_B, \theta_B)$$
$$\theta_B \in [0, \pi]_{mod2\pi},\quad \theta_A \in [i.\pi,\ (i+1).\pi]_{mod2\pi}$$
$$\theta_A = -\theta_B$$

$$FREE\text{: } fixed(A, O_A, \theta_A),\ p\_rotation(B, O_B, \theta_B)$$
$$\theta_B \in [\pi,\ 2\pi]_{mod2\pi},\quad \theta_A = i.\pi_{mod2\pi}$$

$$Transition(MESH, FREE)\text{: } \theta_B = \pi,\ \theta_A = (i+1).\pi$$

3.2 Global Operators

Global operators simplify and abstract behavioral descriptions of complete mechanisms; they are applied to the global region diagram, after the local operators. There are six global operators, most of them generalizations of local operators; their complexity is proportional to the number of regions in the global diagram. Only the first one is compositional.

Kinematic Constraints Simplification: specify which objects can receive input motions, the types of these motions, and their extent (Input-Parts, Input-Types, and Input-Ranges, respectively). Like local kinematic constraints, they are derived from the context in which the mechanism operates. These operators further discard unreachable regions and restrict possible motions in the remaining regions. They can be applied *while* composing local region diagrams, thereby ruling out potential regions without computing them.

Relevance-Set Simplification: filters out behavioral distinctions created by uninteresting objects that introduce too much detail. The user indicates the set of relevant objects whose behaviors s/he is interested in; the most common relevance set is the input/output parts set. This operator projects each region into a new region from which the possible motion labels and relations of uninteresting objects are removed. As a consequence, contiguous regions whose difference lay solely in the behavior of uninteresting objects become qualitatively similar and can thus be merged. This operator is both sound and complete, but *not* compositional.

Region-Difference Abstraction: defines how qualitative behavioral distinctions are made. By relaxing the criteria that establish the difference between two kinematic behaviors, coarser behavioral descriptions are obtained. We consider two relaxations of the qualitative region difference criteria (section 2): (1) motion parameter relations can be *monotonically* identical, instead of identical. Two relations are monotonically identical iff they both specify the same relation (<, =, or >) and their functions are both simultaneously increasing or decreasing in the given interval, and (2) Motion predicates need not be all identical. For example, to distinguish between objects that move and objects that don't regardless of their motion type, we define two categories: "no-motion", consisting of the motion type *fixed,* and "motion", consisting of all the other motion types.

Behavior Parametrization and Periodicity Abstraction: These two operators are direct extensions of the local diagram operators. Regions are parameterized by taking into account all motion labels, motion parameter relations, and intervals. Periodicity abstraction finds isomorphisms between subgraphs by using the extended definition of region similarity.

The order in which the global operators are applied follows the same rationale of local operators: the global kinematic constraints simplifications are applied first during the computation of the global region diagram because they discard potential regions without computing them explicitly. Then, Relevance-Set is applied to focus on the behaviors of relevant objects. Next, Region-Difference is applied to create a coarser behavioral description; if the number of regions does not decrease, these abstractions are unnecessary. Periodicity abstraction is applied at the end.

As an example, consider the effects of global operators on the description the cylinder lock of Fig. 3. We assume that the key can only receive input motions along axis O, and that the only behaviors of interest are those of the key and the cylinder. Applying the kinematic constraints simplifications to the local descriptions of the pairs key/lower pins, (K,Pi), discards all the regions in which *K* and Pi are not in contact, except for the region containing the initial placement. This eliminates the $3^5 — 1$ potential global regions resulting from the pin position combinations when the key is outside the cylinder. This operator also eliminates all the regions where pins not in contact with the key are in different positions as the key is inserted; there are $3x\,(3^4 - f3^3 + 3^2 - f3^1) = 360$ such potential global regions. The Relevance-Set simplification merges all the regions specifying different pin positions but no difference in the angular position of the cylinder or the key. The resulting projected region diagram consists of three regions:

OUT The key and the cylinder are not in contact. The key can both rotate and translate along O, and the cylinder cannot rotate.

IN The key is inside the cylinder. The key translate along O, but cannot rotate. The cylinder cannot rotate either.

UNLOCK The key is at the end of the cylinder. The key can rotate together with the cylinder, but it. cannot translate.

No further application of Region-Difference or Periodicity abstraction is necessary.

4 Comparing Two Mechanisms

Mechanism comparison is an example of a task that cannot be automated without simplifications and abstractions. It consists of determining when two mechanisms can be considered kinematically equivalent. The ability to compare two mechanisms is essential for evaluating design solutions and choosing existing mechanisms for redesign.

To compare two mechanisms, we compare their behaviors, rather than their structure. For example, the two crank mechanisms in Fig. 5 are structurally very different; however, they both transform a continuous rotation into a reciprocating translation. But many of the details of this transformation are different: the number of parts, the motion ratios, etc. Thus, a direct comparison of the mechanisms' region diagrams will also indicate that they are different.

The comparison algorithm uses the simplification and abstraction operators to match the descriptions. First, the region diagrams of both mechanisms are computed, incorporating the kinematic and dynamic constraints. Then, the Relevance-Set operator is applied to discard unnecessary behavioral information. The other operators are then applied in the order indicated in sections 3 and 4. After each application, the region diagrams

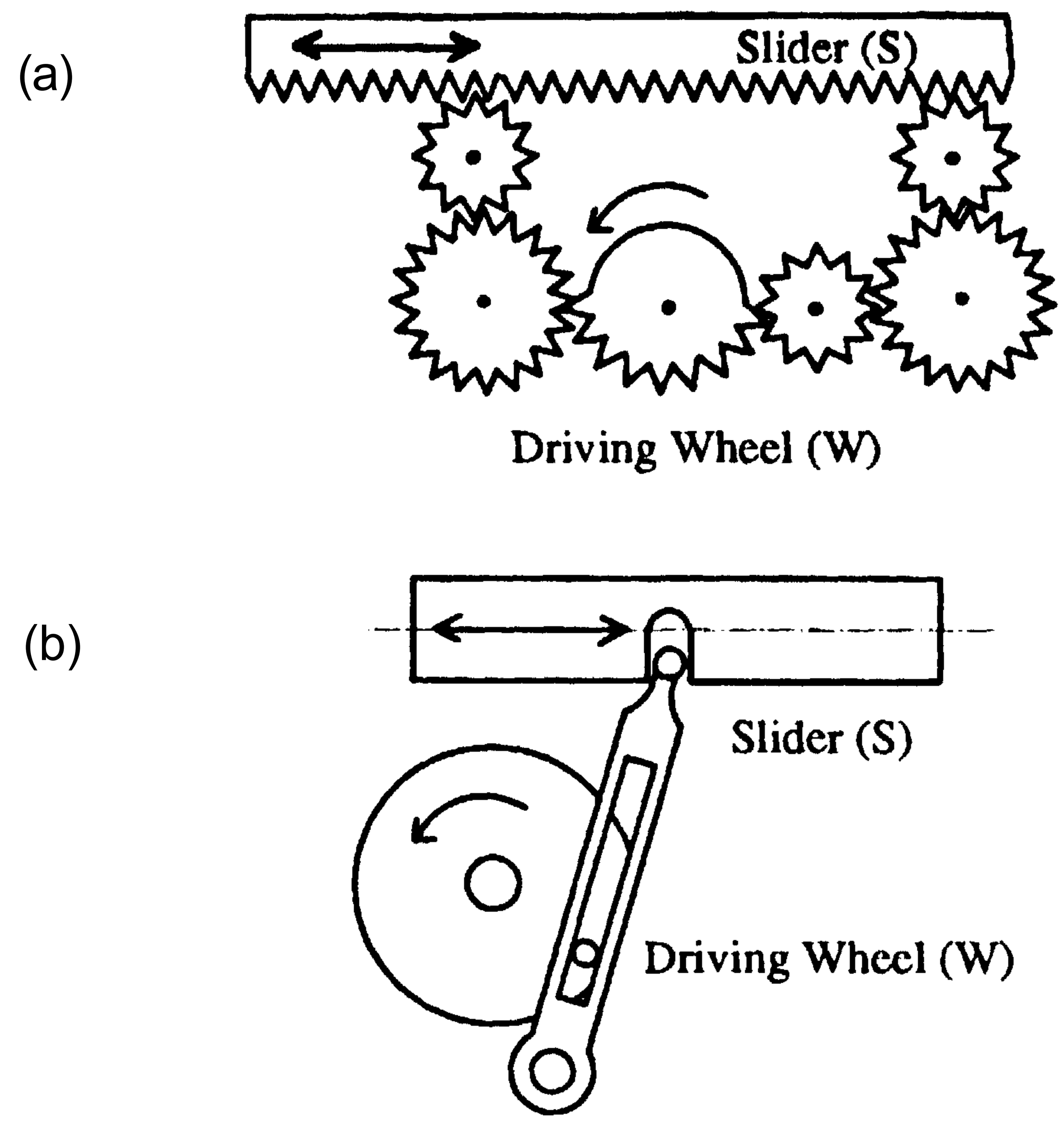

Figure 5: Two Crank Mechanisms

are compared, and a parameterized match between their regions is attempted. When the two diagrams match the algorithm stops, producing a common description for both region diagrams, and the list of operators necessary to achieve the equivalence. This list, establishes the smallest set of conditions necessary to determine when two mechanisms are equivalent (see [Joskowicz, 89b]).

The two crank mechanisms in Fig. 5 can be considered equivalent when: (1) input motions are only applied to the driving wheel; due to friction, the second assembly is not reversible, i.e., the translation of the slider will not cause the rotation of the wheel (kinematic and dynamic constraints) (2) the only behaviors of interest are those of the slider and the driving wheel (Relevance-Set); (3) the motion relations between the driving wheel and the slider are qualitatively similar (Qualitative Abstraction) (4) the gears backlash are negligible (Gap-Closing Abstraction) and (5) both sliders' displacements are inside the desired operating range (Interval Parameterization).

5 Conclusion

Future Intelligent CAD systems must be able to problem-solve using qualitative descriptions at different levels of resolution. This paper presented a set of operators to simplify and abstract kinematic descriptions derived from configuration spaces. To illustrate their use, we showed how mechanism comparison can be done with these operators.

The need for abstracting kinematic descriptions derived from configuration spaces was recognized independently by Nielsen (1988b)[4]. He developed and implemented a number of local abstractions very similar to the ones presented in this paper: Linearization, Qualitative, and Gap-closing abstraction. His treatment of force and motion is similar to our concept of input motion. No global abstractions are proposed.

[4]I am grateful to the anonymous reviewers for bringing this work to my attention.

Producing qualitative descriptions at multiple levels of resolution is an important problem. Recently, [Falkenhainer and Forbus, 88] described a scheme to build large-scale qualitative models; [Murthy, 88] proposed a scheme for dynamically adjusting the level of resolution in which a device is analyzed. Our research shares many of their goals; however, their techniques are inadequate for kinematic behaviors because their qualitative states are defined as n-dimensional rectangular regions of quantity spaces. This distinction is too crude to account for CS-boundaries that partition the space into non-rectangular regions. Moreover, their characterization does not account for different approximations to the exact CS-boundaries.

Possible extensions and future work include: (1) implementing the operators; (2) applying the operators to domains were the problem can be formulated with parameter spaces, e.g., dynamics; (3) exploring the role of the operators in design, and; (4) classifying mechanisms for redesign.

References

[Falkenhainer and Forbus, 88] B. Falkenhainer and K. Forbus, "Setting up Large-Scale Qualitative Models", *Proc. of AAAI-88,* St. Paul, Minnesota, 1988.

[Faltings, 87a] B. Fallings, "Qualitative Place Vocabularies for Mechanisms in Configuration Space", Phd. Diss, UIUCDCS-R-87-1360, U. of Illinois, 1987.

[Faltings, 87b] B. Faltings, "Qualitative Kinematics in Mechanisms" *Proc. of 1JCAI-87,* Milano, Italy, 1987.

[Joskowicz, 87] L. Joskowicz, "Shape and Function in Mechanical Devices", *Proc. AAAI-87,* Seattle, 1987.

[Joskowicz, 88] "Reasoning about Shape and Kinematic Function in Mechanical Devices" PhD. Diss, 402, Courant Institute, New York University, 1988.

[Joskowicz, 89a] L. Joskowicz, "Reasoning about the Kinematics of Mechanical Devices", *Int. Journal of Artificial Intelligence in Engineering,* Vol 4, 1989.

[Joskowicz, 89b] L. Joskowicz, "Mechanism Comparison and Classification for Redesign", IBM Technical Report, to appear, 1989.

[Joskowicz and Addanki, 88] L. Joskowicz and S. Addanki, "From Kinematics to Shape: An Approach to Innovative Design", *Proc. of AAAI-88,* St. Paul, Minnesota, 1988.

[Lozano-Perez, 83] 1 T. Lozano-Perez, "Spatial Planning: A Configuration Space Approach", *IEEE Transactions on Computers,* Vol. C-32, No. 2, 1983.

[Murthy, 88] S. Murthy, "Qualitative Reasoning at Multiple Levels of Resolution", *Proc. of AAAI-88,* St. Paul, Minnesota, 1988.

[Nielsen, 88a] P. Nielsen, "A Qualitative Approach to Mechanical Constraint", *Proc. of AAAI-88,* St. Paul, Minnesota, 1988.

[Nielesen, 88b] P. Nielsen, "A Qualitative Approach to Rigid Body Mechanics", Phd. Diss, UIUCDCS-R-88-1469, U. of Illinois, November 1988.