Qualitative Spatial Representation and Reasoning in Angry Birds: The Extended Rectangle Algebra

Peng Zhang and **Jochen Renz**
Research School of Computer Science
The Australian National University
Canberra, Australia
{u4969566, jochen.renz}@anu.edu.au

Abstract

Angry Birds is a popular video game where the task is to kill pigs protected by a structure composed of different building blocks that observe the laws of physics. The structure can be destroyed by shooting the angry birds at it. The fewer birds we use and the more blocks we destroy, the higher the score. One approach to solve the game is by analysing the structure and identifying its strength and weaknesses. This can then be used to decide where to hit the structure with the birds.

In this paper we use a qualitative spatial reasoning approach for this task. We develop a novel qualitative spatial calculus for representing and analysing the structure. Our calculus allows us to express and evaluate structural properties and rules, and to infer for each building block which of these properties and rules are satisfied. We use this to compute a heuristic value for each block that corresponds to how useful it is to hit that block. We evaluate our approach by comparing the suggested shot with other possible shots.

Introduction

Qualitative spatial representation and reasoning has numerous applications in Artificial Intelligence including robot planning and navigation, interpreting visual inputs and understanding natural language (Cohn and Renz 2008). In recent years, plenty of formalisms for reasoning about space were proposed (Freksa 1992; Frank 1992; Renz and Mitra 2004). An emblematic example is the RCC8 algebra proposed by Randell *et al*. (1992). It represents topological relations between regions such as "x overlaps y", or "x touches y"; however, it is unable to represent direction information such as "x is on the right of y" (Balbiani, Condotta, and del Cerro 1999). The Rectangle Algebra (RA) (Mukerjee and Joe 1990; Balbiani, Condotta, and del Cerro 1999), which is a two-dimensional extension of the Interval Algebra (IA) (Allen 1983), can express orientation relations and at the same time represent topological relations, but only for rectangles. If we want to reason about topology and direction relations of regions with arbitrary shapes, we could for example combine RCC8 and RA, which was analysed by (Liu, Li, and Renz 2009). However, if we only consider the minimum bounding rectangles (MBR) of regions, RA is expressive enough to represent both direction and topology.

What is common in basically all qualitative spatial representations that have been proposed in the literature, and in particular for two-dimensional calculi such as the RA, is that the represented world is assumed to be represented from a birds-eye view, that is from above like a typical map. This is particularly remarkable since qualitative representation and qualitative reasoning are usually motivated as being very close to how humans represent spatial information. But humans typically don't perceive the world from a birds-eye view, the "human-eye view" is to look at the world sideways! When looking at the world sideways, the most important feature of the world is the distinction of "up" and "down" and the concept of gravity. What goes up must come down, everything that is not supported by something and is not stable will invariably fall to the ground. This gives a whole new meaning to the concept of consistency which is typically analysed in the context of qualitative spatial reasoning: a qualitative representation Θ is consistent if there is an actual situation θ that can be accurately represented by Θ. But if any possible θ is not stable and would fall apart under gravity, then Θ cannot be consistent.

In this paper we are concerned with the human-eye view of the world, i.e., we want to be able to have a qualitative representation that takes into account gravity and stability of the represented entities.

Ideally, we want a representation that allows us to infer whether a represented structure will remain stable or whether some parts will move under the influence of gravity or some other forces (e.g. the structure is hit by external objects). Additionally, if the structure is regarded as instable, we want to be able to infer the consequences of the instability, i.e., what is the impact and resulting movement of the instable parts of the structure. Gravity and stability are essential parts of our daily lives and any robot or AI agent that is embedded in the physical world needs to be able to reason about stability. Obvious examples are warehouse robots who need to be able to safely stack items, construction robots whose constructions need to be stable, or general purpose household robots who should be able to wash and safely stack dishes. But in the end, any intelligent agent needs to be able to deal with stability, whenever something is picked up, put down or released. While this problem is clearly of

general interest, the motivating example we use in this paper is the popular Angry Birds game. There we have a two-dimensional snapshot of the world, ironically in the human-eye view and not the birds-eye view, and inferring how objects move and fall down under gravity and other forces is of crucial importance when trying to build an AI agent that can successfully play the game.

The Rectangle Algebra is not expressive enough to reason about the stability or consequences of instability of a structure in a human-eye view. While it is possible to some degree, we could simply impose that a structure can only be consistent if each rectangle is supported from below, this is only a necessary condition, it is clearly not sufficient. For example, in Fig. 1(a) and (b), assume the density of the objects is the same and object 2 is on the ground. The RA relation between object 1 and object 2 in these two figures are both <start inverse, meet inverse>, but obviously the structure in Fig. 1(a) is stable whereas object 1 in (b) will fall. In order to make such distinctions, we need to extend the granularity of RA and introduce new relations that enable us to represent these differences. In this paper, we introduce an *Extended Interval Algebra* (EIA) which contains 27 relations instead of the original 13. We use the new algebra as a basis for an *Extended Rectangle Algebra* (ERA), which is obtained in the same way as the original RA. Depending on the needs of an application, we may not need to extend RA to 27 relations in each dimension. Sometimes we only need the extended relations in one axis. Thus, the extended RA may include 13×27, 27×13 or 27×27 relations depending on the requirement of different tasks.

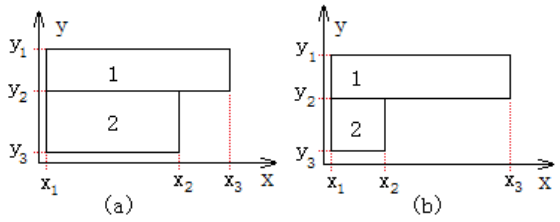


Figure 1: Two structures with the same RA relation (si,mi)

In the following, we will introduce EIA and ERA and then develop a rule-based reasoning method that allows us to infer the stability of a structure and the consequences of the detected instability. Our method is focused towards our target application, namely, to build an agent that can play the Angry Birds game automatically and rationally. This requires us to identify a target that will make a given stable structure maximally instable and will lead to the greatest damage to the structure. In order to do so, we take a stable structure (typically everything we see is by default stable) and for each possible target object we remove the object and analyse the consequences of the structure minus this object. Other applications may require a slightly different method, but the basic stability detection we develop is a general method. The results of our evaluation show that the agent based on this method is able to successfully detect stability and to infer the consequences of instability.

Interval Algebra and Rectangle Algebra

Allen's Interval Algebra defines a set $\mathcal{B}_{int}$ of 13 basic relations between two intervals (see Fig.2). It is an illustrative model for temporal reasoning. The set of all relations of IA is the power set $2^{\mathcal{B}_{int}}$ of $\mathcal{B}_{int}$. The composition ($\circ$) between basic relations in IA is illustrated in the transitivity table in Allen [1983]. The composition between relations in IA is defined as $R \circ S = \cup\{A \circ B : A \in R, B \in S\}$.

Relation	Illustration	Interpretation
X b Y Y a X	X Y	X takes place before Y
X m Y Y mi X	X Y	X meets Y(*i* stands for *inverse*)
X o Y Y oi X	X Y	X overlaps with Y
X s Y Y si X	X Y	X starts Y
X d Y Y di X	X Y	X during Y
X f Y Y fi X	X Y	X finishes Y
X = Y	X Y	X is equal to Y

Figure 2: The 13 basic relations of the Interval Algebra

RA is an extension of IA for reasoning about 2-dimensional space. The basic objects in RA are rectangles whose sides are parallel to the axes of some orthogonal basis in 2-dimensional Euclidean space (see Fig. 1). The basic relations of RA can be denoted as $\mathcal{B}_{rec} = \{(A, B)|A, B \in \mathcal{B}_{int}\}$. The composition between basic RA relations is defined as $(A, B) \circ (C, D) = (A \circ C, B \circ D)$.

The Extended Rectangle Algebra (ERA)

In order to express the stability of a structure and reason about the consequences of the instability in a situation which observes physical rules, we extend the basic relations of IA from 13 relations to 27 relations denoted as $\mathcal{B}_{eint}$ (see Fig.3). The relations consider the importance of the centre of mass of an object for its stability, which corresponds to its centre point for rectangular objects.

Definition 1 (The extended IA relations)**.** *We introduce the centre point of an interval as a new significant point in addition to the the start and end points. For an interval a, denote centre point, start point and end point as c_a, s_a and e_a, respectively.*
1. The 'during' relation has been extended to 'left during', 'centre during' and 'right during' (ld, cd & rd).

- *"x ld y" or "y ldi x" :* $s_x > s_y, e_x \leq c_y$
- *"x cd y" or "y cdi x":* $s_x > s_y, s_x < c_y, e_x > c_y, e_x < e_y$
- *"x rd y" or "y rdi x" :* $s_x \geq c_y, e_x < e_y$

2. The 'overlap' relation has been extended to 'most overlap most', 'most overlap less', 'less overlap most' and 'less overlap less'(mom, mol, lom &lol).

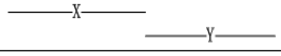
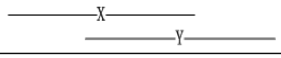
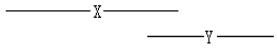
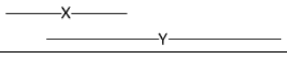
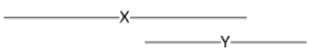
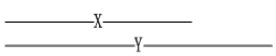
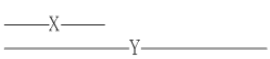
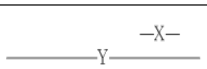
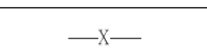
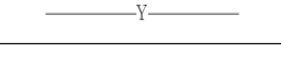
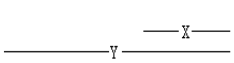
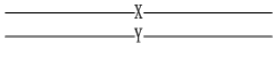
Relation	Illustration	Interpretation
X b Y Y a X		X takes place before Y
X m Y Y mi X		X meets Y (*i* stands for *inverse*)
X mom Y Y momi X		most part of X overlaps with most part of Y
X lol Y Y loli X		less part of X overlaps with less part of Y
X mol Y Y moli X		most part of X overlaps with less part of Y
X lom Y Y lomi X		less part of X overlaps with most part of Y
X ms Y Y msi X		X starts Y and cover most part of Y
X ls Y Y lsi X		X starts Y and cover less part of Y
X ld Y Y ldi X		X during left part of Y
X rd Y Y rdi X		X during right part of Y
X cd Y Y cdi X		X during Y and the midperpendicular of Y through X
X mf Y Y mfi X		X finishes Y and cover most part of Y
X lf Y Y lfi X		X finishes Y and cover less part of Y
X eq Y		X is equal to Y

Figure 3: The 27 basic relations $\mathcal{B}_{eint}$ of the extended IA

- *"x mom y" or "y momi x"* : $s_x < s_y, c_x \geq s_y, e_x \geq c_y, e_x < e_y$
- *"x mol y" or "y moli x"* : $s_x < s_y, c_x \geq s_y, e_x < c_y$
- *"x lom y" or "y lomi x"* : $c_x < s_y, e_x \geq c_y, e_x < e_y$
- *"x lol y" or "y loli x"* : $c_x < s_y, e_x > s_y, e_x < c_y$

3. The 'start' relation has been extended to 'most start' and 'less start' (ms & ls).

- *"x ms y" or "y msi x"* : $s_x = s_y, e_x \geq c_y$
- *"x ls y" or "y lsi x"* : $s_x = s_y, e_x > s_y, e_x < c_y$

4. Similarly, the 'finish' relation has been extended to 'most finish' and 'less finish' (mf & lf).

- *"x mf y" or "y mfi x"* : $s_x > s_y, s_x \leq c_y, e_x = e_y$
- *"x lf y" or "y lfi x"* : $s_x > c_y, s_x < e_y, e_x = e_y$

Denote the set of relations of extended IA as the power set $2^{\mathcal{B}_{eint}}$ of the basic relation set$\mathcal{B}_{eint}$. Denote the set of relations of extended RA as the power set $2^{\mathcal{B}_{erec}}$ of the basic relation set$\mathcal{B}_{erec}$. The composition table of EIA and ERA is straightforward to compute.

Note that EIA can be expressed in terms of INDU relations (Pujari, Kumari, and Sattar 2000) if we split each interval x into two intervals x_1 and x_2 that meet and have equal duration. However, this would make representation of spatial information very awkward and unintuitive. There is also some similarity with Ligozat's general intervals (Ligozat 1991) where intervals are divided into zones. However, the zone division does not consider the centre point.

Inferring stability using ERA rules

In the example in Figure 1 we have seen a simple case where a situation is instable. However, since instability is not a concept which is contained in ERA itself, but is rather a consequence of certain ERA cases, compositional reasoning which is typically used in qualitative spatial reasoning is not helpful for determining instability. Instead, we express standard physical stability rules using ERA relations and use rule-based reasoning to infer stability.

For our analysis we assume that the centre of mass of all objects we represent is at the centre of their MBR. This is always the case for circles and rectangles with a uniform density, even if the rectangles are not parallel to the ground. This covers most objects used in Angry Birds, with the exception of triangular shaped objects and non-structurally significant objects. The MBRs of all Angry Birds objects are obtained using the object recognition module of the basic Angry Birds game playing software that is provided by the organisers of the Angry Birds AI competition (AIBIRDS). In order to get the real shapes of objects, we used an improved version of the object recognition module (Ge et al. 2014), which will be added to the basic software package soon. This module also detects the ground, which is important for stability analysis. In accordance with Ge and Renz (2013) we call rectangles that are equivalent to their MBRs *regular* rectangles and otherwise *angular* rectangles. Figure 4 gives an overview of the different kinds of angular rectangles that Ge and Renz distinguish. The distinctions are mainly whether rectangles are fat or slim, or lean to the left or right. Interestingly, these distinctions can be expressed using ERA relations by comparing the projections of each of the four edges of the angular rectangle with the projections of the corresponding MBR.

We first translate all objects we obtain using the object recognition modules into a set of ERA constraints Θ. Since all these objects are from real Angry Birds scenes, all ERA constraints in Θ are basic relations and the identified objects are a consistent instantiation of Θ. This is a further indication that compositional reasoning is not helpful for determining stability. The next step is that we extract all contact points between the objects. This is possible for us to obtain since we have the real shapes of all objects. The contact points allow us to distinguish different types of contact between two objects, which we define as follows:

Definition 2 (Contact Relation)**.** *Two rectangles R_1 and R_2 can contact in three different ways:*

- *If one corner of R_1 contacts with one edge of R_2, then R_1 has a* point to surface contact *with R_2, denoted as*

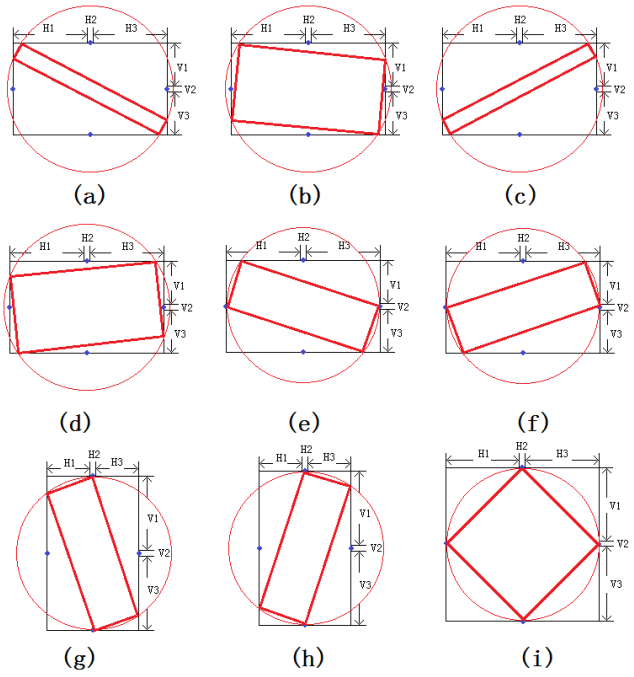


Figure 4: 9 types of angular rectangles (Ge and Renz 2013)

$CR(R_1, R_2) = ps$

- *If one edge of* R_1 *contacts with one edge of* R_2*, then* R_1 *has a* surface to surface contact *with* R_2*, denoted as* $CR(R_1, R_2) = ss$
- *If one corner of* R_1 *contacts with one corner of* R_2*, then* R_1 *has a* point to point contact *with* R_2*, denoted as* $CR(R_1, R_2) = pp$
- *If* R_1 *does not touch* R_2*, then* R_1 *has* no contact *with* R_2*, denoted as* $CR_{R_1,R_2} = n$

Definition 3 (Contact Dimension)**.** *The* contact dimension *expresses in which dimension (horizontal or vertical) two rectangles* R_1 *and* R_2 *contact each other. Specifically, if* $CR(R_1, R_2) \in \{ps, ss, pp\}$*, then:*

- R_1 *and* $_2$ contact horizontally *(denoted as* $CD(R_1, R_2) = hc$*), if* $ERA(R_1, R_2) \in (\{m, mi\}, U_{EIA})$*, where* $\mathcal{U}_{EIA}$ *is the set of all EIA relations.*
- *In all other cases* R_1 *and* R_2 contact vertically*, denoted as* $CD(R_1, R_2) = vc$*.*

If $CR(R_1, R_2) = n$*, then* $CD(R_1, R_2) = n$*, too.*

Stability of Regular Rectangles

We start with stability for regular rectangles as the rules are obvious consequences of Newtonian physics. Throughout this paper, we will never consider what is happening *above* an object when determining its stability. As an example how this can influence stability, refer again to Figure 1(b). Suppose we put a very heavy object 3 on object 1, then object 1 might be stable if the combined centre of mass of objects 1 and 3 is above object 2. Since this requires numerical calculations and cannot be obtained qualitatively, we will always ignore these cases, and consequently will only be able to obtain an approximation of stability. We therefore call our stability concept *ERA-stability*. ERA-stability is based on the simple physical rule that an object is stable and will not topple if the vertical projection of the centre of mass of an object falls into the area of support base.

Given a set of ERA-constraints Θ and a consistent instantiation θ, where the centre of mass of all elements $x \in \theta$ coincides with the centre of the MBR of x. ERA-stability for regular rectangles $x \in \theta$ is determined recursively using the following rules, until no more objects are determined ERA-stable:

A regular rectangle $x \in \theta$ is ERA-stable, if one of the following rule applies:

Rule 1.1: x lies on the ground.

Rule 1.2: $\exists y, z \in \theta$ such that y and z are both ERA-stable and $CD(y, x) = CD(z, x) = vc$: $ERA_x(x, y) \in \{momi, moli, lomi, loli, msi, lsi, ldi\} \wedge ERA_x(x, z) \in \{mom, mol, lom, lol, mfi, lfi, rdi\}$.

Rule 1.3: $\exists y \in \theta$ such that y is ERA-stable and $CD(y, z) = vc$: $ERA_x(x, y) \in \{ms, mf, msi, ls, mfi, lf, cd, cdi, ld, rd, mom, momi, lomi, mol\}$.

Rule 1.4: $\exists y, z \in \theta$ such that y and z are both ERA-stable, $CD(x, y) = hc$ and $CD(x, z) = hc$: $((ERA_x(x, y) \in \{mi\} \wedge ERA_x(x, z) \in \{mom, mol, lom, lol, mfi, lfi, rdi\}) \vee (ERA_x(x, y) \in \{m\} \wedge ERA_x(x, z) \in \{momi, moli, lomi, loli, msi, lsi, ldi\}))$

Rule 1.2 covers the cases where the vertical projection of the mass centre of an object is located between two ERA-stable supporters y and z. Rule 1.3 covers the cases where it is located above the contact area of an ERA-stable supporter y. Rule 1.4 covers cases where x is horizontally contacting with an ERA-stable object on one side and has an ERA-stable supporter on the other side. Fig. 5 shows examples of the four ERA-stable cases for a regular rectangle. In all four rules x is always ERA-stable.

Stability of Angular Rectangles

The case of angular rectangles is more complicated than regular rectangles, but it follows the same physical rule that if an object is stable, the vertical projection of the centre of mass must falls into a support area. While the support area for regular rectangles is defined by two supporting objects, for angular rectangles it can be defined by the ground, acting as a supporting object, and one supporting object. Angular rectangles always have a momentum to become regular rectangles. The rectangles in Figure 4(a,b,e) have a momentum in counter-clockwise direction, the rectangles in Figure 4(c,d,f) have a momentum in clockwise direction. The three rectangles in the bottom row probably never occur in reality as the top point has to be exactly above the bottom point and we ignore these cases. The momentum means that an angular rectangle always needs support to prevent it from becoming a regular rectangle. In cases where the bottom

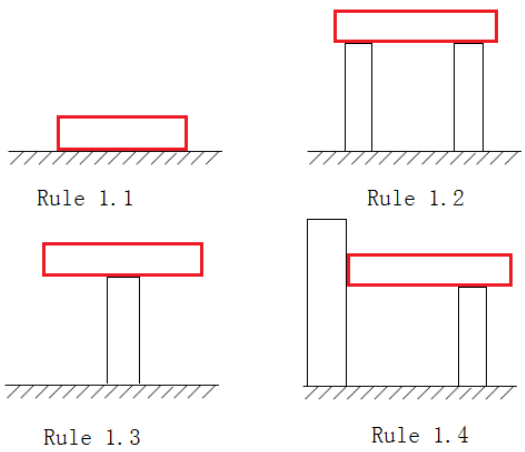


Figure 5: Examples for ERA-stable cases of regular rectangles

point of an angular rectangle is supported, for example by the ground, the second support must always be on the opposite side of the centre of mass. This leads to the following rules, which again are applied recursively, until no more objects are determined ERA-stable:

An angular rectangle $x \in \theta$ is ERA-stable, if one of the following rule applies:

Rule 1.5: $\exists y, z \in \theta$ such that y and z are both ERA-stable and $CD(y,x) = CD(z,x) = vc$: $ERA_x(x,y) \in \{momi, moli, lomi, loli, msi, lsi, ldi\} \wedge ERA_x(x,z) \in \{mom, mol, lom, lol, mfi, lfi, rdi\}$.

Rule 1.6: $\exists y \in \theta$ such that y is ERA-stable and $CD(y,x) = vc$ and $CR(x,y) = ss$: $ERA_x(x,y) \in \{ms, mf, msi, ls, mfi, lf, cd, cdi, ld, rd, mom, momi, lomi, mol\}$.

Rule 1.7: $\exists y, z \in \theta$ such that y and z are both ERA-stable and $CD(y,x) = hc$ and $CD(z,x) = vc$: $((ERA_x(x,y) \in \{mi\} \wedge ERA_x(x,z) \in \{mom, mol, lom, lol, mfi, lfi, rdi\}) \vee (ERA_x(x,y) \in \{m\} \wedge ERA_x(x,z) \in \{momi, moli, lomi, loli, msi, lsi, ldi\}))$.

Rule 1.8: $\exists y, z \in \theta$ such that y and z are both ERA-stable and $CD(y,x) = CD(z,x) = hc : ERA_x(x,y) \in \{m\} \wedge ERA_x(x,z) \in \{mi\}$

The principle of Rule 1.5 is similar to Rule 1.2 which tests if the centre of mass falls between two vertical supporters. Rule 1.6 is similar to Rule 1.3, the difference is that it requires the two objects to be surface-to-surface contact, otherwise the angular rectangle will topple. The supporter in this case also needs sufficient support to remain stationary. Rule 1.7 is a similar rule as Rule 1.4 which describes horizontal support for angular rectangles. Rule 1.8 describes a different property of angular rectangles, which is an angular rectangle can remain stable with the support of two horizontally contacting objects on both left and right sides. This is because the two objects on the sides can provide upward

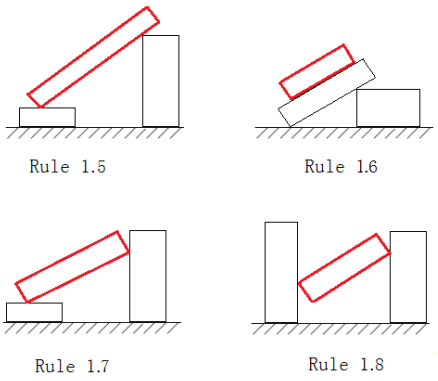


Figure 6: ERA-stable cases of angular rectangles

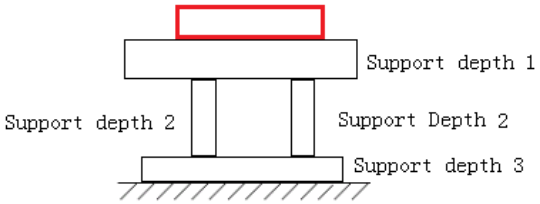


Figure 7: Illustration of support structure

frictions to support the angular rectangle. Fig. 6 shows example cases of the above four rules where angular rectangles are ERA-stable.

All these rules only consider objects as support that are already stable. Cases where two rectangles are stable because they support each other are not considered. Applying rules 1.1-1.8 recursively to all objects in θ until no more objects are detected as ERA-stable identifies all ERA-stable objects. If all objects in θ are ERA-stable, then θ is ERA-stable. The recursive assignment of ERA-stability to objects allows us to build a *stability hierarchy*. Regular objects that lie on the ground are objects of *stability level* 1. Then, objects that are stable because they are supported by an object of stability level ℓ and possible objects objects of stability level smaller than ℓ are considered to be of level $\ell + 1$. Conversely, we could define the *support depth* of a target object (see Figure 7). Those object that are directly responsible for the ERA-stability of a target object x as they support it have support depth 1. Objects that are directly responsible for the ERA-stability of an object y of support depth ℓ as they support it have support depth $\ell + 1$. This can be calculated while recursively determining stability of objects. The support level can be helpful in order to separate the support structure of an object from the larger structure, or when determining which object could cause the most damage to a structure when destroyed.

Applying ERA-stability to identify good shots in Angry Birds

The goal of the Angry Birds game is to shoot birds using a slingshot in a way that all pigs are killed with the fewest number of used birds. The pigs are typically supported by complicated structures that have to be destroyed in order to kill the pigs. Instead of shooting at a pig directly, it might be better to destroy the whole structure that supports the pig. This could be optimised by using the support structure we defined in the previous section.

Another useful substructure is the shelter of the pigs. The reason is straightforward, if a pig is not reachable, there must be some objects that protect it; these objects form the sheltering structure of the pigs. Destroying the sheltering structure can either kill the pig or make the pig directly reachable to the bird. We define the sheltering structure of a pig as the closest objects that protect the pig from a direct hit from each direction. Specifically, a sheltering structure of an object (usually a pig) consist of left, right and top sheltering objects. In order to obtain the sheltering structure of a certain object, the first step is to get the closest object from the left side of the queried object; then, get the supportee list of the object (similar process as getting the supporter list); after that, get the right closest object with its supportee list. The next step is to check if the two supportee lists have objects in common. If so, pick the one with smallest depth as the roof object of the sheltering structure; if not, there is no sheltering structure for the queried object. If a roof object is found, also put the supportees of both the left and right closest objects with smaller depth than the roof object into the sheltering structure. Finally, put the supporters of both left and right closest objects which are not below the queried object into the sheltering structure. This can also be described using rules expressed in ERA (actually RA is sufficient).

The integration of the rules to evaluate a shot

With the rules described above, we are able to dynamically analyse the possible consequences after a shot has been made. In order to predict the final consequence of an external influence on the structure (usually the impact from a bird), the direct consequence and its following subsequences should be analysed in detail. Funt (1987) suggested a similar method to simulate the consequence of a structure with a changed object which assumes that the changed object disappears and chooses the most significant instable object to simulate the consequence.

In the following, we analyse different cases of how a structure can be impacted from an external force and, in particular, what other objects will be affected by an impact to the target object. We separately analyse *regular rectangles* and *angular rectangles*, each of which can be affected in four different ways. First we consider the four cases where the affected objects are regular rectangles:

Case 1: The target object is directly hit by another object. The direct consequence to the hit object will be one of three types: **destroyed, toppling, remaining stationary**. Empirically, the way to determine the consequence of the hit depends on the height and width ratio of the target.

Algorithm 1 Direct hit estimation

```
1:  PROCEDURE Hit(Object o)
2:  o ∈ pendingList
3:  if o ∈ affectedList then
4:      pendingList.remove(o)
5:      Exit()
6:  end if
7:  if o.isRegular then
8:      if o.affectedMethod = "Hit" then
9:          if o.isDestroyed then
10:             affectedList.add(o)
11:             pendingList.add(supporteesOf(o))
12:         else if !o.isDestroyed then
13:             if o.height/o.width > threshold then
14:                 affectedList.add(o)
15:                 pendingList.add(supporteesOf(o))
16:                 pendingList.add(supportersOf(o))
17:             else
18:                 pendingList.add(supportersOf(o))
19:             end if
20:         end if
21:     end if
22: end if
23: pendingList.remove(o)
```

For example, if an object hits a target with the height and width ratio larger than a certain number (such as 2), the target will fall down. And this ratio can be changed to determine the conservative degree of the system. In other words, if the ratio is high, the system tends to be conservative because many hits will have no influence on the target. Moreover, if the external object affects a target with the height and width ratio less than one, the target itself will remain stable temporarily because the system will also evaluate its supporter to determine the final status of the target. After deciding the direct consequence of the hit, the system should be able to suggest further consequences of the status change of the direct target. Specifically, if the target is destroyed, only its supportees will be affected. If the target falls down, the case will be more complex because it may influence its supporters due to the friction, as well as its supportees and neighbours. If the target remains stable temporarily, it will also influence its supporters, and its supporters may again affect its behaviour. Algorithm 1 demonstrates the evaluation process of an object that has been hit by another object.

Case 2: The supportee of the target object topples down. The target object's stability is also determined by its height to width ratio, but the number should be larger (about 5) as the influence from a supportee is much weaker than from a direct hit. If the target is considered as instable, it will fall down and affect is neighbours and supporters; otherwise, it will only influence its supporters (see Fig. 8).

Case 3: The supporter of the target object topples down.

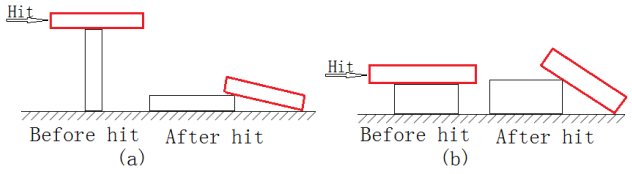


Figure 8: Two examples for case 2.

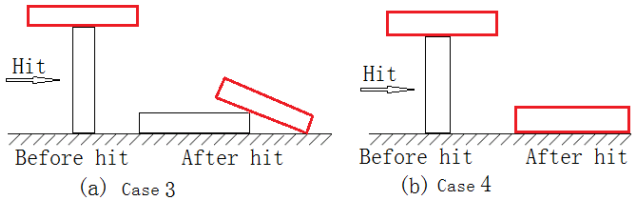


Figure 9: Examples for Cases 3&4

Here a structure stability check process (applying Rules 1.1 - 1.4) is necessary because after a supporter falls, the target may have some other supporters and if the projection of its mass centre falls into the areas of its other supporters, it can stay stable. Then, if the target remains stable, it will again only affect its supporters due to the friction; otherwise, it may fall and affect its supporters, supportees and neighbours (see Fig. 9(a)).

Case 4: The supporter of the target is destroyed. This is more like a sub case of the previous one. If the target cannot remain stable after its supporter is destroyed, it may fall and affect its supporters, supportees and neighbours (see Fig. 9(b)).

Next we consider the four cases where the target objects are angular rectangles:

Case 5: The target object is directly hit. As analysed in Section *Inferring stability using ERA rules*, angular rectangles can be classified into 9 classes. If we only consider the direction of the objects, there only exists 3 classes, which are *"lean to left"*,*"lean to right"* and *"stand neutrally"*. Assume the hit is from left to right. Then if a *"lean-to-left"* object is hit and the force is strong enough, the object will rotate around the lowest pivot, i.e. the lowest corner of the object. However, in the Angry Birds game, before the force becomes sufficient to make the object rotate, the object will be destroyed first. Thus, the two possibilities here are either the target object affects its supporters or it is destroyed and affects its supportees. The case of *"lean-to-right"* object is the same as above. For a *"stand-neutrally"* object, it cannot stand by itself and at least one *point-to-surface* touched supporter on each side is necessary. If it is hit from left, there will be no friction applied to its left supporter, thus it will either be destroyed or affect its right supporter.

Case 6: The supportee of the target object topples down. Angular objects will not topple due to the fall of their supportees. Because their supporters restrict their spatial location. And therefore only the supporters will be affected. For *"stand-neutrally"* objects, if the friction given by their supportees is from left to right, the left supporters will not be affected, and vice-versa for the right supporter.

Case 7: The supporter of the target object topples down. First check the stability using Rules 1.5 - 1.8. If the object still has sufficient support, it only affects its supporters, otherwise it will rotate and affect its supportees, other supporters and neighbours.

Case 8: The supporter of the target object is destroyed. Again, a stability check is applied first. However if it remains stable, it will not affect its supporters because no friction applies to the target from the destroyed supporter. If it is not stable, than it will affect its supportees, other supporters and neighbours.

By applying these cases recursively for each potential target object, we obtain a list of objects that are affected when hitting a particular target object. Then, with all the affected objects in a list, the quality of the shot can be evaluated by calculating a total heuristic value depending on the affected objects. The scoring method is defined as follows and is based on experimenting with different heuristic values: if an affected object belongs to the support structure or the sheltering structure of a pig, 1 point will be added to this shot; and if the affected object is itself a pig, 10 points will be added to the shot. After assigning scores to shots at different possible target objects, the target with the highest score is expected to have the largest influence on the structures containing pigs when it is hit. The heuristic value roughly corresponds to the number of affected objects plus a higher value for destroyed pigs. Then, based on different strategies, the agent can choose either to hit the reachable object with highest heuristic value or generate a sequence of shots in order to hit the essential support object of the structure.

We first extract the ERA relations between all objects and then match the rules for all relevant combinations of objects. Thus, the process of evaluating the significance of the targets is straightforward and fast.

Planning in Angry Birds

The previous analysis only looks at the effects of one shot, while taking into account reachability of target objects. Reachability is calculated using the impact different bird types have on different block types. For example, a yellow bird can hit through 2 layers of wood, a blue bird can go through 2 layers of ice, etc. We can use this knowledge to reason about the reachability of a bird and roughly estimate the damage of the direct consequence of a shot. Ideally, we would like to plan a sequence of shots. Due to the inability to accurately predict the state after a shot (we can currently only predict affected objects), successful planning of shot sequences seems very hard.

However, in order to demonstrate how planning in Angry Birds could work, we have implemented one special case of planning, which can be used when the shot with highest score is not reachable due to some blocks in the path. We

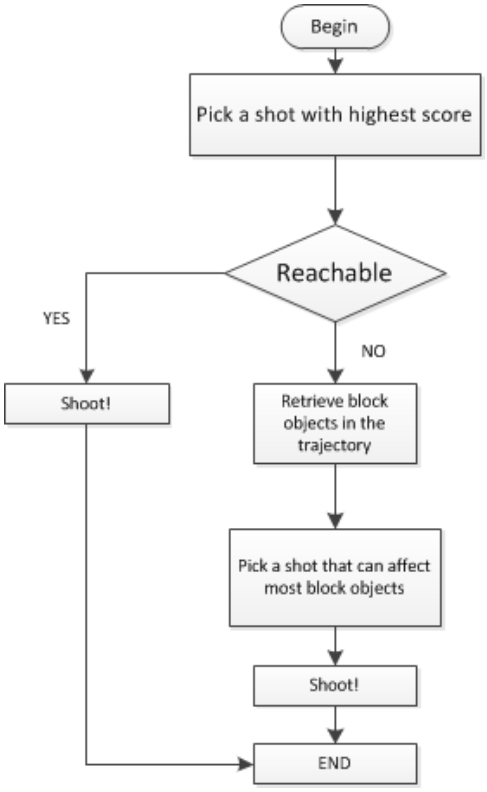


Figure 10: Process of the proposed two step planner

Table 1: Evaluation on first 21 levels. #RO is the number of reachable objects, S1, S2, and S3 are the scores obtained when shooting at the highest, second-highest and third highest ranked object. HS is the rank of the reachable object with the highest score.

Level	#RO	S1	S2	S3	Time(s)	HS
1	6	29030	30370	27330	1.34	2
2	7	16240	6040	11250	1.06	1
3	5	5410	41850	6790	0.77	4
4	2	36890	1850		1.45	1
5	6	22380	66240	16800	5.11	2
6	10	1610	6770	7710	1.83	4
7	10	46200	6230	6210	3.81	1
8	7	17310	17310	6010	1.08	4
9	9	14030	10140	13210	5.37	1
10	9	22920	3860	5920	3.88	1
11	10	57110	22640	9680	6.60	1
12	6	26570	15010	21800	3.47	1
13	8	12200	12650	14460	4.02	3
14	10	30330	28590	37670	2.43	7
15	9	16880	17550	4580	3.97	2
16	11	15090	22430	23860	4.24	3
17	12	48850	12240	19850	7.18	1
18	10	9460	14590	3240	3.02	2
19	2	5870	3290		5.83	1
20	8	7420	8170	6820	4.46	2
21	8	14270	4650	7170	12.70	1

can identify which blocks are on the trajectory and block the target. Then a planner could be used to generate a shot to first clear the blocks in the path and then hit the primary target with the second shot. Specifically, the planner will first check if the highest-ranked shot is reachable. If not, it will retrieve the block objects in the calculated trajectory. Then it will search for a shot that can affect most of the objects in the block list. Finally, the planner will suggest using the first shot to clear the blocks if applicable. However, as we cannot exactly simulate the consequence of a shot, sometimes the first shot may lead to an even worse situation. Initial experiments we did with this method showed good results though. Fig.10 shows the process of the planner.

Evaluation

We evaluate our approach by applying it to the first 21 Poached Eggs levels, which are freely available at chrome.angrybirds.com. These levels are also used to benchmark the participants of the annual Angry Birds AI competition. We take each level and calculate the heuristic value of all reachable objects and rank them according to the heuristic value. We then fire the bird at each of the reachable objects and record the actual game score resulting from each shot. We reload the level before each shot, so each bird always fires at the initial structure. We restricted our analysis to initial shots only, as we found that it is not possible to consistently obtain the same intermediate game state simply by repeating the same shot. However, our method works equally well for intermediate game states.

The results of our evaluation are shown in Table 1 where we listed for each of the 21 levels how many reachable objects there are, the resulting game scores of shooting at the four highest ranked reachable objects, as well as the rank of the object that received the highest game score. It turns out that in about half of the levels (10 out of 21), our method correctly identified the best object to hit. In a further 5 cases, the second highest ranked object received the highest score, in 2 cases the 3rd highest ranked object. On average there are about 7.8 reachable objects. The average score of the shooting at the highest ranked object over all 21 levels is about 21,700, shooting at the second ranked object gives an average of about 16,700 and the third ranked object an average of about 13,100. This clearly demonstrates the usefulness of our method in identifying target objects.

However, the resulting game score is not always a good measure, as shots can have unexpected side effects that lead to higher scores than expected. These side effects are typically unrelated to the stability of a structure and, hence, are not detected by our method. As an example, consider level 16 where the second and third ranked objects resulted in a higher score than the first ranked object. Figure 11 shows the level and output of our stability analysis. Our method predicts that shooting at object 30 will kill the two pigs on the left and result in the toppling of the left half structure.

Figure 12 shows the actual result after the recommended shot has been made. We can see that the prediction of the

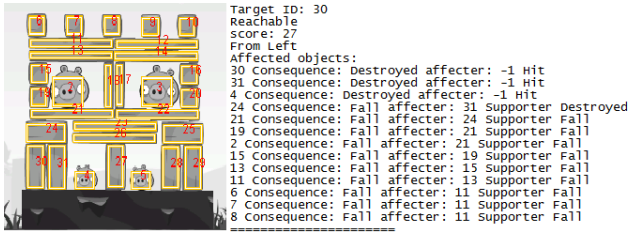


Figure 11: Level 1-16 in Chrome Angry Birds and agent output for this level

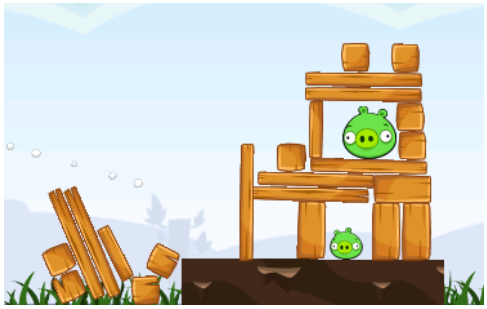


Figure 12: Actual result for the recommended shot

system is very accurate in this case. The left two pigs are killed and the left part of the structure toppled. However, the shots at the second and third ranked objects each killed three pigs and therefore obtained higher scores. It turns out that two of the three pigs were killed when they bounced to the wall without actually destroying much of the structure (see Figure 13). Therefore, the next shot is much less likely to solve the level as more of the structure remains. So the shot at the highest ranked object first is likely still the best shot overall as the second shot can easily solve the level. This example nicely shows that Angry Birds requires even more sophisticated methods to correctly predict the outcome of shots.

Figure 13: Actual result for the 3rd ranked shots

Related Work

Qualitative spatial representation and reasoning has been applied to some physical systems to do common sense reasoning (Klenk et al. 2005). Physical reasoning has two important theories, namely kinematics and dynamics(Forbus 1988). Kinematics mainly concerns about the position of objects which may change continuously over time and the shape which is strictly rigid and static over time (Davis 2008). However it considers less about the forces between objects and the type of motions. These features make kinematics easy to be formulated, on the other hand models using kinematics are usually limited by the context and appear to be less expressive. The CLOCK Project (Forbus, Nielsen, and Faltings 1991) is a typical success which uses a variety of techniques but mainly kinematics to represent and reason about the inner motions of a mechanical clock. Under some given environments and constraints, this approach can successfully analyse the mechanisms; however, as this system requires restricted constraints such as the exact shape of the parts and the range of the motion, it may not be applicable in the cases with high degrees of freedom such as the world of Angry Birds.

In contrast, dynamics takes force analysis into account which allows reasoning about more complex motions and the transfer of motion. However, the limitation is that precisely predicting a consequence of a motion in an arbitrary environment is almost impossible due to uncertainty. However, it is possible to reason about some simpler physical features of a structure, such as stability. Fahlman (1973) implemented a stability verification system dealing with the stability of blocks with simple shapes (e.g. cuboid). This system can produce qualitative output to determine whether a structure is stable without external forces, but it requires some quantitative inputs (e.g. magnitude of forces and moments). The stability check in our work is partially inspired by this system, however without quantitative inputs about forces, we use a purely qualitative way to analyse the probability of the balance of forces and moments in a structure. Although our approach may produce a few false predictions in very complex cases, it is a good estimation of humans' reasoning approach. Our work also focuses on reasoning about the impacts of external forces (a hit from an external object) on a structure, which has not been discussed much in previous work.

Additionally, a method for mapping quantitative physical parameters (e.g. mass, velocity and force) to different qualitative magnitudes will be very useful for the situation where the input information is often incomplete. (Raiman 1991) developed a formalization to define and solve *order of magnitude equations* by introducing comparison operators such as $\approx$ (in the same magnitude) and $\ll$ (far less than). Raiman's method is of particular significance for our approach, for example, we will be able to infer that a hit from 1×1 block will not change the motion state of a 100×100 block with the same density.

Discussion and Future Work

In this paper we have introduced an extended rectangle algebra useful for representing and reasoning about stability under gravity and other properties of 2-dimensional structures. By splitting some basic interval relations into more detailed ones, we obtained 27 interval relations in each dimension that can express the physical relations between rectangular

objects more precisely. Our algebra is very useful when representing a human-eye view of the physical world, rather than the birds-eye view which is surprisingly common in qualitative spatial representation and reasoning. A nice example of a two-dimensional human-eye view of the world is Angry Birds, where we can represent the world using the new algebra.

In addition to representing what we see using the extended interval algebra, we also used it for defining some useful structural rules regarding properties such as stability, reachability, support, and shelter that allow us to perform rule-based reasoning about these properties. We tested the usefulness of our rules by designing an agent that performs a structural analysis of Angry Birds levels. Based on these rules, we predict for each block the consequences if it gets hit and calculate a heuristic value that determines the usefulness to hit the block. We then shoot at the block with the highest value that is reachable with the current bird. Our evaluation shows that our method is very successful in determining good target objects and good shots.

Despite its success, some parts of our method offer significant room for improvement. For example, our current motion analysis can only vaguely predict which objects may be affected by a shot, i.e., we do not have accurate predictions about a possible successor state at this stage. Because of this, we currently cannot plan ahead and evaluate sequences of shots in advance. Also we only consider stability of an object with respect to its supporters, but not with respect to objects that are on top of it, which means we do not analyse the interactions between substructures. In the future, we will try to design a feedback loop between the dynamic analysis and ERA, specifically, so we can use the results of the dynamic analysis to restrict a range of possible subsequent positions of an object and use ERA rules to suggest several possible successor cases. Then we can use dynamic analysis again to check the stability of these cases and finally decide on desirable successor states. Moreover, we will find a method to determine useful substructures in order to increase the reliability of the dynamic analysis.

Acknowledgements

This research was supported under Australian Research Council's Future Fellowship and Discovery Projects funding schemes (project numbers FT0991917 and DP120104159).

References

AIBIRDS. Angry Birds AI competition. http://aibirds.org.

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26:832–843.

Balbiani, P.; Condotta, J.; and del Cerro, L. 1999. A new tractable subclass of the rectangle algebra. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 442–447.

Cohn, A. G., and Renz, J. 2008. Qualitative spatial representation and reasoning. In *Handbook of Knowledge Representation*. Oxford, England: Elsevier. 551–584.

Davis, E. 2008. Physical reasoning. In *Handbook of Knowledge Representation*. Oxford, England: Elsevier. 597–620.

Fahlman, S. E. 1973. A planning system for robot construction tasks. Technical report, Cambridge, MA, USA.

Forbus, K. D.; Nielsen, P.; and Faltings, B. 1991. Qualitative spatial reasoning: the clock project. *Artificial Intelligence* 51(1-3):417–471.

Forbus, K. D. 1988. Commonsense physics: a review. *Annual Review of Computer Science* 3(1):197–232.

Frank, A. U. 1992. Qualitative spatial reasoning about distances and directions in geographic space. *Journal of Visual Languages & Computing* 3(4):343–371.

Freksa, C. 1992. Using orientation information for qualitative spatial reasoning. In *Theories and Methods of SpatioTemporal Reasoning in Geographic Space*.

Funt, B. V. 1987. Problem-solving with diagrammatic representations. In *Readings in computer vision: issues, problems, principles, and paradigms*, 456–470. Morgan Kaufmann.

Ge, X., and Renz, J. 2013. Representation and reasoning about general solid rectangles. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 905–911.

Ge, X.; Gould, S.; Renz, J.; Abeyasinghe, S.; Keys, J.; Wang, A.; and Zhang, P. 2014. Angry Birds game playing software version 1.3: Basic game playing software. http://www.aibirds.org/basic-game-playing-software.html.

Klenk, M.; Forbus, K.; Tomai, E.; Kim, H.; and Kyckelhahn, B. 2005. Solving everyday physical reasoning problems by analogy using sketches. In *Proceedings of the 20th National Conference on Artificial Intelligence*, 209–215.

Ligozat, G. 1991. On generalized interval calculi. In *Proceedings of the 9th National Conference on Artificial Intelligence*, 234–240.

Liu, W.; Li, S.; and Renz, J. 2009. Combining RCC-8 with qualitative direction calculi: Algorithms and complexity. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 854–859.

Mukerjee, A., and Joe, G. 1990. A qualitative model for space. In *Proceedings of the 8th National Conference on Artificial Intelligence*, 721–727.

Pujari, A. K.; Kumari, G. V.; and Sattar, A. 2000. Indu: An interval duration network. In *Proceedings of the 16th Australian joint conference on AI*, 291–303.

Raiman, O. 1991. Order of magnitude reasoning. *Artificial Intelligence* 51(1):11–38.

Randell; Cui, Z.; and Cohn, A. G. 1992. An interval logic for space based on connection. In *Proceedings of the 10th European Conference on Artificial Intelligence*, 394–398.

Renz, J., and Mitra, D. 2004. Qualitative direction calculi with arbitrary granularity. In *Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence*, 65–74.