Regular Equivalence and Dynamic Logic

Maarten Marx[*]and Michael Masuch
Institure for Logic, Language and Computation,
Department of Sociology, Universiteit van Amsterdam
The Netherlands
{marx,masuch}@science.uva.nl

March 28, 2002

Abstract

This paper describes a precise linguistic counterpart to the notion of a regular equivalence relation on a social network. That is, a formal language of position terms is defined with the property that on finite networks, two actors are regularly equivalent if and only if they cannot be distinguished by a position term. The paper also contains an exact characterization of the set of complex relations which are preserved under regular equivalences.

The results presented here are known from logic and computer science, in which the mentioned language is called dynamic logic. The aim of the paper is to make these results available to social network analysts and explain why they are of interest to them.

Regular equivalence is an elegant and well known tool to define social positions. It has been studied quite intensively within social network analysis. This paper describes a precise linguistic counterpart to the model–theoretic notion of regular equivalence. That is, a formal language of position terms is defined with the property that on finite networks,

> two actors are regularly equivalent if and only if they cannot be distinguished by a position term.

[*]Research supported by NWO grants 612–062–001 and 612-000-106

This formal language is well known in the fields of logic and computer science under the name of dynamic logic [4]. The notion of regular equivalence was discovered independently (from each other and from the social science literature) in these two fields and proved extremely useful.

The goal of the paper is to make the key results on regular equivalence which are obtained in logic and in computer science known and available to the social network community. The language of position terms is valuable for social network analysts because it provides deep insight in the meaning of regular equivalence as a formalization of the notion of social position. In particular, it gives information on the *scope* of the formalization. The formal language describes exactly which social positions are definable in which terms. It can also serve as a valuable *tool*. This tool can be used for instance to look for evidence against a claim that two actors occupy the same social position.

Plan of paper. The paper is organized as follows. We start with briefly looking at social positions, networks and regular equivalence and their historical developments in the fields of social network analysis, logic and computer science. The following section introduces regular equivalence formally and relates it to the formation of block models. The main results of the paper are in Sections 3 and 4. Readers familiar with regular equivalence may safely start reading here. In Section 3, the language of position terms is defined and the key results about it are stated and explained. Section 4 is about complex relations. Complex relations are formed from primitive ones using algebraic operations: e.g., the ancestorhood relation is the transitive and reflexive closure of the parenthood relation. It answers the question: exactly which complex relations are preserved under regular equivalence? The paper finishes with a number of concluding remarks.

1 Social positions, networks and regular equivalence

The notions of social position and social role are key concepts in social network analysis. The standard view is that a social position is a property of social actors. Thus given a set of actors or a social network, a position is most naturally defined as a subset of actors: those who have that property.

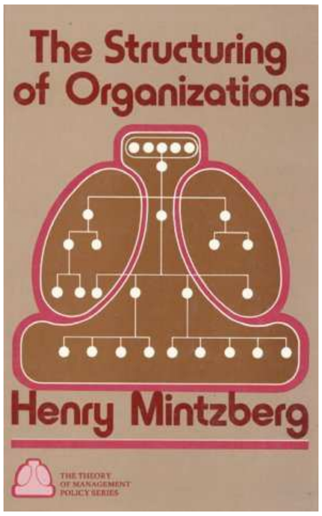

Figure 1: Mintzberg's famous five positions in an organization.

What determines a position? According to Wasserman and Faust,

> Subsets of actors with similar roles are equivalent, and occupy the same network position. [16] p.465

There are several ways to formalize this idea of position. But whatever choice one makes, the basic technical setup is the same. Given a social network, positions in that network should be determined by an equivalence relation on the set of actors. The equivalence classes then give the different positions in a network. For a famous example, see Figure 1, the cover of Mintzberg [8]. There are several proposals in the literature for equivalence relations; we mention *structural equivalence* [7], *regular equivalence* [13, 17] and *local role equivalence* [18]. [10] relates these and more proposals (a deeper analysis is in her book [11]), cf. also [16].

In this paper, we will solely concentrate on regular equivalence as the relation which defines social positions. Thus we will study the following formalization of social position,

(1) two actors occupy the same social position if and only if they are regularly equivalent.

The definition of regular equivalence is quite involved, but in one line it can be summarized as

(2) Actors who are regularly equivalent have identical ties to and from regularly equivalent actors.

A precise definition will be given below. The technical definition can be traced back to graph–theory [6] and to automata theory [14] (late 60's). Around the same time (late 70's, early 80's) the notion of regular equivalence arose independently in social network analysis [13, 17], modal logic [1], and in computer science [9, 5]. In modal logic, the relation used to be called zig–zag relation, in computer science it is called bisimulation, and that name stuck.

Networks occur under the name of *Kripke models* in philosophical logic and as *labeled transition systems (LTS)* in the computer science literature. LTS's are used to model processes. Nodes in a network are the states in some process and the (labeled) ties are transitions between states caused by performing a certain action (indicated by the label of the tie). If two nodes s and t are related —in their jargon, *bisimilar*— then every process starting at s can be simulated when starting at t and vice–verse. Thus from this perspective the two states are equivalent or indistinguishable. In modal logic, the notion of bisimulation has a comparable function but came from a different motivation. It's main use is in establishing the precise expressive power of a formal language.

Since it's incubation, the notion of bisimulation has been studied intensively (8030 hits on Google, 99 for "regular equivalence"). The fields of logic and computer science have met and exchanged ideas very fruitfully. A good example is the proceedings of the workshop "Three days of bisimulation" held at the Dutch center for Mathematics and Computer Science in 1994 [12].

All in all, it seems that regular equivalence is a very robust notion. It is invented independently in (at least) three different areas for more or less similar purposes. It turned out extremely useful in computer science and in logic. And despite its difficult definition it is a natural notion with a very nice and clean logic behind it.

Networks. For most of the paper we will be concerned with networks with a single relation in which the actors may have attributes. Such a network is represented as a structure $\langle A, R, \mathbf{P}\rangle$, in which A is a set of actors, $R \subseteq$

$A \times A$ is a binary relation on the set of actors, and $\mathbf{P}$ is a set of attributes $P_1, P_2, \ldots$, all of which denote a subset of A. Networks are finite if not explicitly stated otherwise. Networks with multiple relations will be discussed too; in that case the network contains a set $\mathbf{R}$ of binary relations on A. We often present networks with pictures, as in Figure 3. In this figure, the actors are represented by letters, the ties by arrows, and the attributes by words or letters attached as a label to the actors. Thus in Network I in that figure, d is a male actor, which receives a tie from b only.

2 Regular equivalence and bisimulation

Within the theory of social networks, the notion of regular equivalence is used to partition a network into sets of actors who occupy the same social position. It is thus an equivalence relation on the set of actors of one network. In computer science and modal logic, the corresponding notion of a bisimulation is mainly used to indicate that two states in *two different networks* are indistinguishable (or equivalent). Here it is thus a relation between two different sets of actors. The second perspective is followed in this paper. This has certain advantages: (1) the first perspective is just a special case, (2) it provides a clear relation between a model and its associated block model, and (3) the more general perspective can be used to compare different (block)models based on the same data. These three advantages will be illustrated below.

Instead of regular equivalence we will use the term bisimulation. The intuition behind it is easily explained. Let x and y be actors in a network. We say that x and y are *bisimilar* if x and y have the same atomic attributes and have ties to bisimilar actors. Figure 2 contains an example of two networks with a bisimulation $B = \{(1, a), (2, b), (2, c), (3, d), (4, e), (5, e)\}$. In this figure, p and q denote atomic attributes.

Definition 2.1 (Bisimulation) *Let $\langle A, R, \mathbf{P}\rangle$ and $\langle A', R', \mathbf{P}'\rangle$ be networks. A non–empty relation $\overset{\leftrightarrow}{\sim} \subseteq A \times A'$ is called a* bisimulation *if it satisfies the following:*

Whenever $x \overset{\leftrightarrow}{\sim} y$ (pronounce: "x and y are bisimilar"), then

(atom) *x and y have the same atomic attributes;*

(forth) *for all ties xRx', there is a tie yRy' and $x' \overset{\leftrightarrow}{\sim} y'$;*

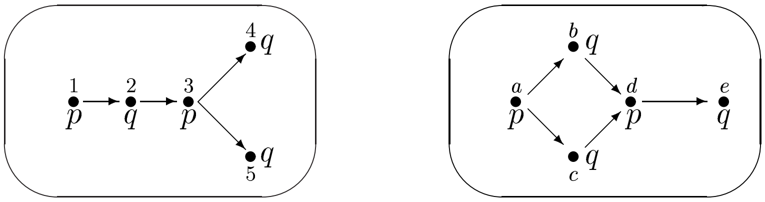

Figure 2: $\{(1,a),\ (2,b),\ (2,c),\ (3,d),\ (4,e),\ (5,e)\}$ is a bisimulation.

back *for all ties yRy', there is a tie xRx' and $x' \overset{\leftrightarrow}{\sim} y'$.*

We note that if $\overset{\leftrightarrow}{\sim}$ is a relation on one network it is an equivalence relation.

Bisimulation captures precisely the out degree part of the definition of regular equivalence given in [17]. [10] argues convincingly for two separate notions of regular equivalence, one for out- and one for in-degree ties. We will use the above notion of bisimulation (and thus of regular equivalence) in Section 3. It is conceptually simpler and all the relevant ideas can already be explained. The generalization to multiple ties and to incoming relations is straightforward and discussed in Section 3.2. We note that the relation in Figure 2 preserves both in and out degree ties.

A well known example is provided in Figure 3. Besides ties there are also attributes in the networks, here only indicated for the actors at the bottom. Let B be the relation between the domain of Network I and that of Network II, indicated in the picture by agents having the same name with or without a quote-mark. That is, $B = \{(a,a'),(b,bc'),(c,bc'),(d,d'),(e,e')\}$. B is not a bisimulation. For instance, because the outgoing tie from actor bc' to a male actor cannot be matched by actor c. In fact, as the reader can easily verify, no bisimulation is possible linking a and a'. In (1) we agree to say that actors occupy the same position if and only if they are regularly equivalent. Now bisimulation is another term for regular equivalence. Thus a and a' occupy *different* positions in the two networks. Fine, but the immediate next question will be "how" or "in what" they differ. Of course, we can point to the two networks and explain the difference in structure: in this way we give a structural or semantical explanation. It would be nice though if we could also give a less abstract, more linguistical explanation. Of course we can. Let R be the verb associated with the tie. The two networks show the subtle but real difference between the following two statements

(3) actor a' Rs an actor who Rs a male *and* a female.

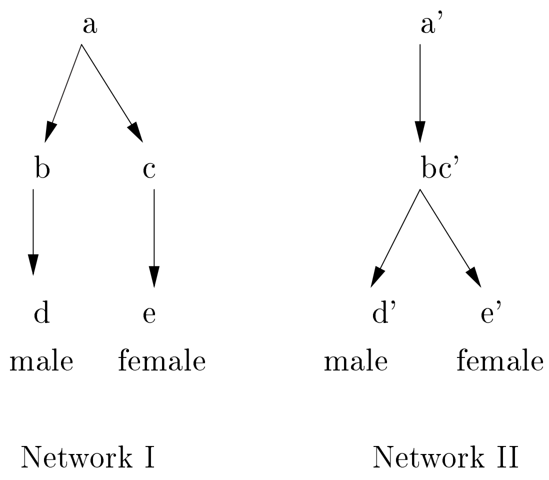

Figure 3: Two networks which do not bisimulate

(4) actor a Rs an actor who Rs a male *and* an actor who Rs a female.

Statement (4) is strictly weaker than (3). In network I, (3) is false for a, and in Network II, (3) is true for a'. This is a much more concrete and useful answer to the question about the difference between a and a'.

This paper is about the relationship between the *semantical* notion of bisimulation or regular equivalence and the *syntactical* or *linguistical* notion of position.

Block models. If the bisimulation relation is between one and the same model, it is precisely the notion of (out degree) regular equivalence. As well known, then the relation partitions the network into sets of bisimilar (or regularly equivalent) actors. These sets are called social positions. The equivalence class consisting of all actors bisimilar to actor a is denoted by $\bar{a}$. It is convenient to turn this set of positions itself into a network. The resulting structure is called a block model. It is constructed as follows.

Definition 2.2 *Let $\langle A, R, \mathbf{P} \rangle$ be a network and $\overset{\leftrightarrow}{\sim}$ a bisimulation on $A \times A$. The block model associated to $\langle A, R, \mathbf{P} \rangle$ and $\overset{\leftrightarrow}{\sim}$ is the network $\langle A^*, R^*, \mathbf{P}^* \rangle$ in* *which,*

$$\begin{array}{lcl} A^* & = & \{\bar{a} \mid a \in A\} \\ \bar{a}R^*\bar{b} & \iff & \textit{there exists } a' \in \bar{a},\ b' \in \bar{b} \textit{ such that } a'Rb' \\ \multicolumn{3}{l}{\bar{a} \textit{ has attribute } P_i^* \iff a \textit{ has attribute } P_i.} \end{array}$$

Note that the third line is well defined since bisimilar actors have the same attributes. The models $\langle A, R, \mathbf{P}\rangle$ and $\langle A^*, R^*, \mathbf{P}^*\rangle$ are of course closely related. With the present machinery this can be made precise in an elegant way.

Proposition 2.3 *The function $f : A \longrightarrow A^*$ defined $f(a) = \bar{a}$ is a bisimulation between $\langle A, R, \mathbf{P}\rangle$ and $\langle A^*, R^*, \mathbf{P}^*\rangle$.*

Proof. We have to show that the conditions of Definition 2.1 hold. By definition, a and $f(a)$ have the same attributes. For the forth-condition, suppose Rab. Then $R^*\bar{a}\,\bar{b}$ by definition of the block model. Whence $R^*f(a)f(b)$, as desired. For the back-condition, suppose $R^*f(a)\bar{b}$, for some $\bar{b}$. We have to find some actor x such that Rax and $f(x) = \bar{b}$. By definition of R^*, there are $a' \in \bar{a}$, $b' \in \bar{b}$ such that $Ra'b'$. Because a and a' are bisimilar, this means that there is a b'' bisimilar to b' and Rab''. As $f(b'') = \bar{b}$, b'' is the actor we are looking for. This finishes the proof. QED

Functional bisimulations are called regular network homomorphisms in [17]. The forth-condition of a functional bisimulation turns into the homomorphism condition: xRy implies $f(x)R'f(y)$.

Different views on the same data. Now that we defined bisimulation as a relation between actors in (possibly) different models, we obtain the possibility to compare different models of the same data set, as in Figure 4. Two different models have been created from one set of network data. Bisimulation provides a way to compare the different models. If for instance an actor a in Model 1 is bisimilar to itself in Model 2, then for that actor the two models agree on his social position. Another use is to compare different block models of the same data set.

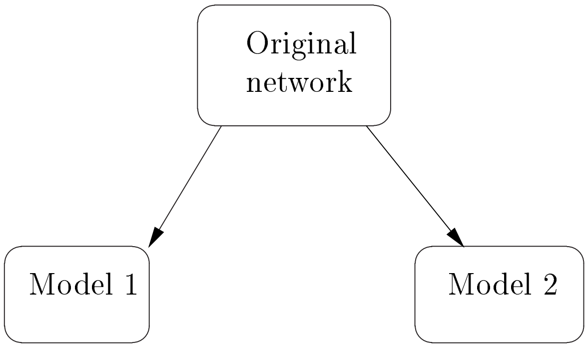

Figure 4: Two models of one data set.

3 A language for defining social positions

In this section a formal language for describing social positions is introduced. This language is well–known in (philosophical and mathematical) logic and in computer science. We present three key results about the relation between this language and the notion of regular equivalence. The language for defining positions naturally leads to a new definition of social position: two actors occupy the same social position if all position terms in the language apply to both of them in the same way. The three key results together imply that this definition is equivalent to the one given in (1). The analysis here thus sheds new light on the definition of social position and provides a valuable new tool for position analysis.

Let a network $\langle A, R, \mathbf{P} \rangle$ be given. A (social) position is a subset of A consisting of actors who are regularly equivalent. Examples of social positions are BOSS, MANAGER, etc. Some positions can naturally be decomposed into more primitive positions by means of set theoretical operations. For example, the set of all mothers is just the intersection of the set of all parents and the set of all females. Within the set of parents, the fathers are those who are not mothers. For our purposes, FEMALE can be considered a primitive notion. (We hesitate to call it a position, it is better described as an attribute of actors. But note that semantically we cannot differentiate between positions and attributes: both are just subsets of the set of actors.) PARENT however is not primitive, it consists of those actors who have at least one child, that is who stand in the has–child relation to some (other) actor.

Between these definitions there is a difference in logical structure. To define MOTHER and FATHER, the Boolean connectives *and* and *not* were sufficient. These definitions are given here at different levels of abstraction:

x is a MOTHER	iff	x is a PARENT **AND** x is a FEMALE
$x \in$ MOTHER	iff	$x \in$ PARENT **AND** $x \in$ FEMALE
MOTHER	$=$	PARENT $\cap$ FEMALE
FATHER	$=$	PARENT $\cap$ **NOT** FEMALE.

The definition of PARENT involved quantification ("there is someone") and a binary relation ("has–child"). Obviously for some social positions it is not only important that someone is related to some other actor, but that he is related to certain actors in certain social positions. For instance, a gay is a man who is sexually attracted to other man.

From these examples, the following requirements for a formal language to describe positions can be distilled.

1. it should incorporate primitive attribute names;
2. the language should be closed under Boolean operations like *and, or, not*;
3. the language can express that an actor is related to an actor in a certain position.

With these desiderata in mind, we propose the following definition. Let **P** be a set of attribute names and let **R** be a set of relation names. The language of *position terms* is defined as follows:

1. every attribute name in **P** is a position term;
2. if P and Q are position terms, then also **NOT** P, P **AND** Q and P **OR** Q are position terms;
3. if P is a position term, and R a relation name, then both $\exists\, \mathsf{R}\, P$ and $\forall\, \mathsf{R}\, P$ are position terms;
4. Only the terms defined in clauses 1–3 are position terms.

Position terms can be used to *define* positions in a network. More precisely, position terms P are used to state that in a network, actor a occupies position P. We denote this by a is P. The meaning of the Boolean connectives is self explanatory. Thus a is PARENT **AND** MALE means that a is a parent and a male, in other words a father. The interesting part comes with the relations. Given a network $(A, R, \mathbf{P})$, we say that a is $\exists\ \mathsf{R}\, P$ if and only if there exists an actor $b \in A$ with an R–tie from a to b and b is P. For example, a is in position $\exists$ has$-$child MALE precisely if a has a child b which is a male. Shortly, if a has a son. The position term $\forall\ \mathsf{R}\, P$ is the dual of $\exists\ \mathsf{R}\, P$. For instance, a is $\forall$ has$-$child MALE if a only has sons. Formally, a is in position $\forall\ \mathsf{R}\, P$ if and only if for all actors $b \in A$ with an R–tie from a to b, b is P. Note that by this definition, a is $\forall$ has$-$child MALE if a has no children at all.

We can define interesting complex positions in this language. For instance, reading "loves" as "is sexually attracted to",

HETERO = (MALE **AND** $\forall$ loves FEMALE) **OR** (FEMALE **AND** $\forall$ loves MALE).

As indicated above, someone can be a hetero according to this definition without being sexually attracted to anybody. Let us call someone who does a happy hetero. It is a hetero who is sexually attracted to some person. As persons are male or female, we can define this as:

HAPPY-HETERO = HETERO **AND** $\exists$ loves (MALE **OR** FEMALE).

In this language we can also express the difference between the two actors a and a' in Figure 3, indicated by the example sentences (3) and (4). The positions expressed in (3) and (4) can be formalized as

(3′) $\exists\ \mathsf{R} \exists\ \mathsf{R}$ (MALE **AND** FEMALE)
(4′) $\exists\ \mathsf{R}$ ($\exists\ \mathsf{R}$ MALE **AND** $\exists\ \mathsf{R}$ FEMALE).

In Network II, a' is $\exists\ \mathsf{R} \exists\ \mathsf{R}$ (MALE **AND** FEMALE) but in Network I a is not in position $3'$. In fact we can strengthen the description of a' in Network II to

a' is $\forall\ \mathsf{R} \exists\ \mathsf{R}$ (MALE **AND** FEMALE),

in natural language, "everybody a' Rs, Rs a male and a female. Obviously this is false for a in Network I, for instance because b does not R a female.

So the language can be used to define specific social positions in networks. But is it really useful to a network analyst? We think it does because it opens a new perspective on the formalization of social position by means of regular equivalence. For one thing, it permits a novel definition of social position, quite different from (1). Let a, b be actors in some network. We say that

(5) a and b occupy the same social position if and only if a and b cannot be distinguished by a position term,

whereby we mean that there is no position term P such that a is P and b is **NOT** P.

Now we have two definitions of the same concept social position. One is structural, the second is syntactical. How are they related?

3.1 Perfect Harmony

Recall the formalization of social position from (1)

> actors a and b occupy the same social position if and only if they are regularly equivalent.

Recall that bisimilar was the computer science synonym for regular equivalent. In the next three key results, regular equivalence is meant as in Definition 2.1: only one relation and only out–degrees. The corresponding language of position terms contains only one relation R. All results generalize to in and out degree regular equivalence with multiple relations. This is discussed in the next subsection.

We will now discuss three fundamental results relating the language of position terms to regular equivalence. References are given in the historical remarks at the end of this subsection. The first result states that the language is well-designed:

Key Result 1 *The language of position terms is well designed.*

> *If a and b are regularly equivalent, then they cannot be distinguished by a position term.*

This result guarantees us that if two actors occupy the same social position according to the structural definition (1), they also occupy the same position according to the syntactical definition (5). We note that the contrapositive is useful in deciding when two actors can *not* be regularly equivalent: just find a position term on which they disagree. This can clarify matters and make things precise. It is simpler to give a single position term on which two actors disagree then to show that for every possible regular equivalence on the network, the actors are not equivalent.

The next key result states the other direction of the first result. Thus together they imply that the structural and linguistical definition of social position are equivalent. As the first key result said that the language was well designed, this result says that it is in a certain sense complete.

Key Result 2 *For finite networks, the language of position terms is complete.*

> *If a and b are* not *regularly equivalent, then they* can *be distinguished by a position term.*

Reformulated, it states that in order to decide whether two actors are regularly equivalent (that is, have the same social position) it is sufficient to check whether they agree on all position terms. We stress that the result holds for finite networks, but fails for infinite ones.

The third key result is concerned with the descriptive power of the positions terms. These are designed to define sets of actors who occupy the same social position. Then the next question is whether *all* of these sets can be defined by position terms. Again the answer is positive, at least when restricted to first order definable[1]sets.

Key Result 3 *The language of position terms is descriptively complete.*

> *Every first order definable set of regular equivalent actors is definable by a position term.*

This result can be understood as follows. If one has decided that social position means regular equivalent, then it is not needed to use full first order logic to define social positions. The simple language of position terms is sufficient for that.

So we can conclude that regular equivalence and the language of position terms are really two sides of the same coin. One cannot be accepted without the other. The language provides the tool to make simple but very precise statements about social positions. In the next section we switch attention from the positions to the ties.

[1] A set is first order definable if it can be defined by a formula from first order predicate logic. This is a very strong notion. For instance, all axioms of ZF set theory are first order formulas.

Historical notes. The language of position terms is a modal language in disguise. For a modern introduction to modal logic see [2]. This also contains full proofs of the three key results. We urge the reader to try to prove the first key result. The proof is by an induction on the complexity of position terms. The two other key results are much harder to prove. The second is due to Hennessy and Milner [5], the third to van Benthem [1].

Modal logic was traditionally concerned with questions regarding necessity, knowledge, belief and temporal aspects of language. Nowadays modal logic is seen as a very versatile knowledge representation language. The present paper is another example of its versatility. Under the name of description logic, very powerful computer programs have been developed for reasoning about position terms. These programs can check for instance whether one position is subsumed by the other (as parent subsumes father), or whether a theory of position terms (like a theory of kinship relations) is consistent. We believe these computational tools can be useful to network analysts. An excellent entry to this field is provided by the description logic web page `http://dl.kr.org`.

3.2 In and out degrees and multiple relations

The original definition of regular equivalence asks for matching in and out going ties. For bisimulation one only considers out going ties. The definitions are easily adjusted. Call a bisimulation which in addition satisfies the forth condition in both directions for the converse of R an *in-out bisimulation*. To witness this stronger structural notion of social position we add two operators $\exists\ \mathsf{R}^{-1}$ and $\forall\ \mathsf{R}^{-1}$ to the language of position terms. They are just the backward looking analogues of $\exists\ \mathsf{R}$ and $\forall\ \mathsf{R}$. Thus a is $\exists\ \mathsf{R}^{-1}\ P$ if and only if there exists some actor b with an R–tie from b to a and b is P. With this addition to the language the three key results go through when regular equivalence is interpreted as in–out bisimulation.

It is also easy to incorporate multiple relations into the definitions. For $\mathbf{R}$ a set of relations, we say that a relation B is an $\mathbf{R}$ in-out bisimulation (Notation: $\overset{\leftrightarrow}{\sim}_{\mathbf{R}}$) if it is an in-out bisimulation for every $R \in \mathbf{R}$. Naturally all three key results also go through when multiple relations are considered.

4 The relations

This section studies which kind of complex relations are preserved under regular equivalence. The relationships between network morphisms and semigroup morphisms have been explored rather well, see Pattison [11] for an overview. That composition of relations is preserved under regular equivalence is known within the social networks literature (in fact that was explicit in the definition of maximal regular equivalence). In this section, we consider all regular expressions and boolean operations on relations and present a concise characterization of all complex relations which are preserved under regular equivalence.

In this section, regular equivalence is synonymous for **R** in-out bisimulation defined in Section 3.2. Thus we study the multiple relations, attribute–valued regular equivalence from [17]. Again, let us see what regular equivalence formalizes:

> The notion of regular equivalence formalizes the observation that actors who occupy the same social position relate in the same ways with other actors who are themselves in the same position [16], (p.473).

In other words, *the relations* to and from actors in (other) social positions determine one's social position. But which relations should then be taken into account? The definition of regular equivalence only mentions primitive relations and their inverses, but from these many other natural relations can be build. The prime example is of course the composition of two relations which comes back in the study of semigroups. But there are many other natural operations on relations, like union or transitive closure. We could even consider all the operations of relation algebra.

The question we address here is one of design. We wonder —given the definition of regular equivalence which only asks for preservation of primitive relations and inverses— which operations on relations are preserved as well by this definition.

It turns out that several natural, most prominently composition, operations on relations are preserved by regular equivalence. We also present some examples of relations which are not preserved. The final conclusion of this analysis will be that the definition of regular equivalence was a very good choice: it takes a large and natural class of relations into account. Though large, the class is generated by five simple operations. We introduce them here formally one by one and illustrate their use by an example of kinship relations. Then the main binary relation will be the “parent of” relation

$$(x, y) \in \textsf{parent-of} \text{ if and only if } x \text{ is a parent of } y.$$

Instead of the cumbersome notation $(x, y) \in \textsf{parent-of}$, we write $x \,\textsf{parent-of}\, y$ and similarly for the other relations.

As relations are in general directed, the operation which turns the order of the arguments around is useful:

$$\textit{inverse} \qquad R^{-1} = \{(x, y) \mid (y, x) \in R\}.$$

Inverses often have a name of their own, for instance $\textsf{parent-of}^{-1}$ is the $\textsf{child-of}$ relation. The next example is widely studied in social network analysis:

$$\textit{composition} \qquad R \circ S = \{(x, y) \mid \text{ for some } z,\ (x, z) \in R \text{ and } (z, y) \in S\}.$$

For instance, the grandparent relation is the composition of two parent relations:

$$\textsf{grandparent-of} = \textsf{parent-of} \circ \textsf{parent-of}.$$

We show that composition of relations is preserved by regular equivalence. Thus suppose that there is a regular equivalence $\overset{\leftrightarrow}{\sim}_{\mathbf{R}}$ on some network and $x \overset{\leftrightarrow}{\sim}_{\mathbf{R}} x'$ holds. We show that the forth-condition holds for the composition $R \circ S$ for any $R, S \in \mathbf{R}$. Thus suppose $xR \circ Sy$. Then there exists a z such that xRz and zSy. Because $x \overset{\leftrightarrow}{\sim}_{\mathbf{R}} x'$ and the forth-condition holds for R, there exists a z' such that $x'Rz'$ and $z \overset{\leftrightarrow}{\sim}_{\mathbf{R}} z'$. Because $z \overset{\leftrightarrow}{\sim}_{\mathbf{R}} z'$ and zSy, again by the forth-condition, there must also be a y' such that $y \overset{\leftrightarrow}{\sim}_{\mathbf{R}} y'$ and $z'Sy'$. Thus, there exists a y' $y \overset{\leftrightarrow}{\sim}_{\mathbf{R}} y'$ and $x'R \circ Sy'$. The back-condition is shown by a symmetric argument.

The following three operations are also preserved. The reader is invited to supply the proofs along the lines just presented.

$$\begin{array}{lrcl}
\textit{union} & R \cup S & = & \{(x, y) \mid (x, y) \in R \text{ or } (x, y) \in S\} \\
\textit{counter domain} & \sim R & = & \{(x, x) \mid \text{ for no } z,\ (x, z) \in R\} \\
\textit{class identity} & P? & = & \{(x, x) \mid x \text{ has attribute } P\}.
\end{array}$$

The parent relation itself can be seen as the union of the father and the mother relation: $\textsf{parent-of} = \textsf{father-of} \cup \textsf{mother-of}$. The operation of counter domain provides a set of individuals; the counter domain of the parent relation is the set of all childless people: $\sim$parent-of is the set of all (x, x) such that x does not have a child.

The operation ? turns an attribute into a relation. This is very useful if the network also contains attributes of actors. For instance in kinship relations, the gender of actors plays an important role. Now we can define father-of as male? $\circ$ parent-of. The name class–identity is taken from [17] (Definition 14).

Even infinitary operations can be preserved under regular equivalence. A nice example is transitive closure: R^+ is the set of all (x, y) such that there exists a finite R–path from x to y. We can view R^+ as the infinite union $R \cup (R \circ R) \cup (R \circ R \circ R) \cup \ldots$ Also for transitive closure there are natural examples in kinship relations, for instance the ancestor relation is the transitive closure of the parent relation: parent-of $^+=$ ancestor-of.

The next proposition states that all relations constructed with these operations are preserved under regular equivalences. The proof is by an induction on the complexity of the relations and follows similar reasoning as given above for composition.

Proposition 4.1 *Let $\overset{\leftrightarrow}{\sim}_{\mathbf{R}}$ be a regular equivalence relation on some network with attributes $\mathbf{P}$ and with relations $\mathbf{R}$. Let S be any relation constructed from the relations in $\mathbf{R}$ and the attributes in $\mathbf{P}$ by means of converse, composition, union, counterdomain, class–identity or transitive closure.*

Then $\overset{\leftrightarrow}{\sim}_{\mathbf{R}}$ is a regular equivalence for the relation S as well.

We note that S may be arbitrary complex and may contain all the operations any number of times.

We now look at two operations which are in general not preserved by regular equivalence.

$$\begin{array}{lccl} \textit{intersection} & R \cap S & = & \{(x, y) \mid (x, y) \in R \text{ and } (x, y) \in S\} \\ \textit{negation} & \bar{R} & = & \{(x, y) \mid (x, y) \notin R\}. \end{array}$$

It is more difficult to find examples in natural language of these relations. The incestuous relation between a parent and a child is an example of an intersection between two relations, namely between the parent-of and the has-sex-with relation. Network (A) in Figure 5 shows two equivalent actors, a and c. But a has an incestuous relation (with d), while b has not (here p stands for parent-of and s for has-sex-with). Thus intersection is in general not preserved under regular equivalence. Network (B) shows that also negation of relations is not preserved; here a and b are equivalent, b stands not in a relation to c, but a stands in a relation to all who are equivalent to c.

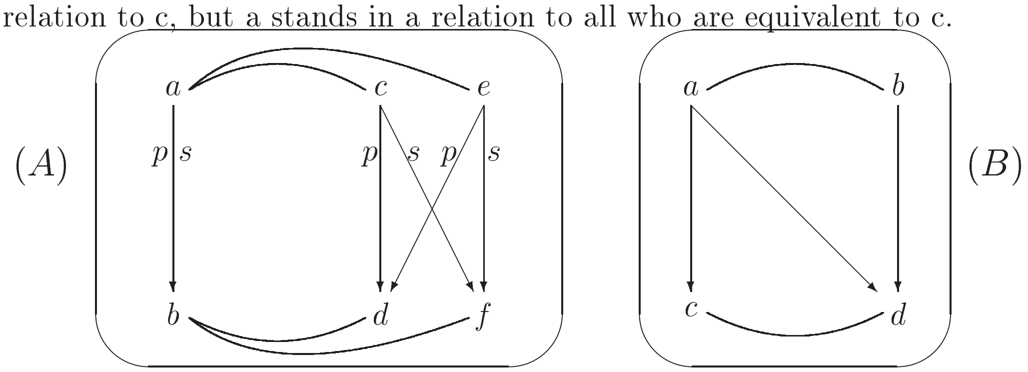

Figure 5: Counterexamples for intersection and negation

Having seen examples of operations which are preserved and which are not, it is natural to ask a *completeness* question: can we give a list of operations on relations such that every relation which is preserved under regular equivalence can be expressed by operations from this list?

It turns out that if we restrict ourselves to *first order definable* relations, the list from Proposition 4.1 without the second order transitive closure is complete (van Benthem, [15], Theorem 5.17).

Key Result 4 *A first order definable relation is preserved under* **R** *in-out regular equivalence if and only if it can be defined from the relations in* **R** *using inverse, composition, union, counterdomain, and class-identity.*

For second order definable relations including the transitive closure operator the question of completeness remains open.

5 Conclusion

We have defined a formal language of position terms with the property that on finite networks,

> two actors are regularly equivalent if and only if they cannot be distinguished by a position term.

Because of this completeness property the language is adequate to describe social positions defined as regular equivalence classes. We believe this is a useful new perspective. We hasten to remark that such an analysis is not restricted to regular equivalence. In principle, mathematical logic has the tools to define linguistic counterparts for every notion of equivalence on networks. For some notions (like regular equivalence) this analysis has been done, for others this represents challenging research problems, in particular for the interesting modification to regular equivalence by Boyd and Everett [3].

Our second contribution concerned the preservation of complex relations under regular equivalence. Within social network analysis these kind of questions have been restricted to compositions of relations [11, 17]. We have looked at all regular expressions and presented a concise characterization of all complex relations which are preserved under regular equivalence.

We hope this paper can be a catalyst to a fruitful exchange of ideas between the fields of logic and social network analysis. Both can benefit enormously from each other. Logic can provide a number of technical results and several potentially useful computational tools. Conversely there seems to be much more to gain from a traffic of ideas. We mention one example domain in which the theory of social networks has a lot to offer to both logic and artificial intelligence. This is the field of agent technology, which is concerned with networks of multiple agents which have knowledge, beliefs, desires, intentions, etc. Here there are many difficult puzzles and paradoxes, like the muddy children puzzle. More importantly, real world applications are at present being constructed, which consist of networks of interacting high–level artificial agents. These applications are used for instance in electronic auctions and in airplane–traffic control systems.

References

[1] J. van Benthem. *Modal Correspondence Theory.* PhD thesis, Mathematisch Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam, 1976.

[2] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, 2001.

[3] J. Boyd and M. Everett. Relations, residuals, regular interiors, and relative regular equivalence. *Social Networks*, 21:147–165, 1999.

[4] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.

[5] M. Hennessy and R. Milner. Algebraic laws for indeterminism and concurrency. *Journal of the ACM*, 32:137–162, 1985.

[6] P. J. Higgins. *Notes on Categories and Groupoids*. van Nostrand Reinhold, London., 1971.

[7] F. Lorrain and H. C. White. The structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1:49–80, 1971.

[8] H. Mintzberg. *Structuring of Organizations*. Prentice Hall, 1978.

[9] D. Park. Concurrency and automata on infinite sequences. In *Proceedings 5th GI Conference*, pages 167–183. Springer, 1981.

[10] P. Pattison. Network models: Some comments on papers in this special issue. *Social Networks*, 10:383–411, 1988.

[11] P. Pattison. *Algebraic Models for Social Networks*. Cambridge University Press, Cambridge, UK, 1993.

[12] A. Ponse, M. de Rijke, and Y. Venema, editors. *Modal Logic and Process Algebra: A Bisimulation Perspective*, volume 53 of *Lecture Notes*. CSLI Publications, 1995.

[13] L. D. Sailer. Structural equivalence: Meaning and definition, computation and application. *Social Networks*, 1:73–90, 1978.

[14] P. H. Starke. *Abstract Automata*. North Holland, 1972.

[15] J. van Benthem. *Exploring Logical Dynamics*. Studies in Logic, Language and Information. CSLI Publications, Stanford, 1996.

[16] S. Wassermann and K. Faust. *Social Network Analysis*. Cambridge University Press, Cambridge, UK, 1994.

[17] D. R. White and K. P. Reitz. Graph and semigroup homomorphisms on netwoks and relations. *Social Networks*, 5:193–234, 1983.

[18] C. Winship and M. Mandel. Roles and positions: A critique and extension of the blockmodeling approach. In Samuel Leinhardt, editor, *Sociological Methodology 1983-1984*, pages 314–344. Jossey-Bass, 1983.