The Human Automaton

Justen Rickert

May 5, 2017

Contents

1 Purpose

The ultimate purpose of this paper is to derive a fashion in which a machine, or `automaton`, may derive the meaning behind, or understand, human language sentences. A general sentence notwithstanding context does not mean anything, yet humans have somehow come to be able to throw sentences around at each other in such a way that their conversations are meaningful. That is to say that people have come to develop a rapport with each other wherein seemingly-random phrases of a speaker, even when devoid of `context`, somehow are meaningful to, or verifiable by, a listener. The penultimate purpose of this paper is therefore to derive a construction of `context` by a which a machine may use to begin to understand the seemingly-random phrases of a speaker speaking to it.

2 Language Construction

A mathematical framework for language construction is important to the derivation of the `computational model` to be defined. In particular the theory of mathematical logical and the theory of computation happen to be very important to this language construction.

There will be a concatenation of two varieties of languages in mathematics used to describe the way in which `natural`—that is to say *human*—`languages` and `contexts` can be modeled. To clarify what is meant by `context` here, imagine the statement "I am sad because my dog died." It may be clear to the reader why sadness can be determined by the possibility of the dog dying. It is probably because the reader has either himself had a dog, or has at some point seen or understood the joy that having a dog can bring to the person (perhaps the reader has seen or read *Where the Red Fern Grows*). Imagine, however, a culture that never came to the point of domesticating dogs. A person of such a culture reading such a statement would not understand the interconnected nature that exists between *being sad* and *having a dog die*. A person who understands the statement has the `context` available to him to understand the logic of why *a dog dying* could imply *being sad*. In any case, the derivation of this logic will be described according to the logic of a mathematical `first-order language`.

2.1 First-order Language

Let's first consider a mathematical language of logic, a `first-order language` consisting of a finite collection of sentence symbols, say $P, P_1, \ldots, P_n$ or $Q, Q_1, \ldots, Q_n$; the negation symbol, $\neg$; and the symbol for implication, $\rightarrow$. A single sentence symbol is considered a valid statement. The negation symbol, $\neg$, acts upon one sentence symbol, while the implication symbol, $\rightarrow$, joins together two sentence symbols. The following are all valid statements in the defined `first-order language`:

- P
- $\neg P$
- $P \rightarrow Q$

If P is considered to be the statement "It is raining," then the negation of the statement, *i.e.* $\neg P$, becomes the statement "It is not raining." When the statement Q is considered to be "The ground is wet outside," the implication of the two statements, *i.e.* $P \to Q$, becomes "If it is raining, then the ground is wet outside." When the statement P holds, meaning when it *is* raining outside, it must be that the statement Q must hold as well, as it makes sense in the `natural language`—English, in this particular case—that the statement $P \to Q$ holds, as it would be impossible for the ground *not* to be wet outside if it is raining. So when the statement P holds and the statement $P \to Q$ holds, then statement P derives statement Q, written $P \vdash Q$.

Statements constructed more complexly than this require parenthesizing, otherwise the statements may be read ambiguously. Consider the ambiguous statement $P_1 \to P_2 \to P_3$. Let P_1 to be "Somebody is at the door," let P_2 to be "The dog is barking at the door," and let P_3 to be "The man is angry at the dog." There are two different ways to parenthesize this statement, each having a different effect on the underlying meaning of the statement. They are $P_1 \to (P_2 \to P_3)$, or $(P_1 \to P_2) \to P_3$.

Consider $P_2 \vdash P_3$ holds, or "The dog is barking at the door" and "If the dog is barking at the door, then the man is angry at the dog," so that it can be derived that P_3 holds, as it makes sense that the man is angry because the dog is barking. This corresponds to a parenthesized statement $P_1 \to (P_2 \to P_3)$, where P_2 holds. In this construction the complete derivation of the statement does not make complete sense from a `natural language` perspective because the statement P_1, the somebody being at the door, does not have any implication on the scenario according to the definition of implication in the `first-order language`. The dog has already been defined to be barking at the door, and there has been no logical implication that there is somebody at the door. The dog may just be barking at the door for no reason.

The other parenthesized statement $(P_1 \to P_2) \to P_3$ means something different. Consider $P_1 \vdash P_2$ holds, or "Somebody is at the door" holds and "If somebody is at the door, then the dog is barking at the door" holds, so that P_2 is derivable. As the dog must be barking at the somebody who is at the door, then it must be that the dog is barking at the door. Furthermore since $(P_1 \to P_2)$ holds, and it is that the complete statement $(P_1 \to P_2) \to P_3$ holds, as this complete statement makes sense given that the man must be mad because the dog is barking at the door, then it must be that $(P_1 \to P_2) \vdash P_3$.

In either case, the point is that the two statements considered by their different parenthesis constructions carry along with them different meaning (even without the consideration that either $P_1 \vdash P_2$ holds or $P_2 \vdash P_3$ holds). At the heart of this `first-order language` are the theorems of soundness and completeness for propositional logic. The theorem of soundness means that given a properly-constructed sentence, as previously talked about, there is a sound derivation of the sense that the statement is trying to convey. The theorem for completeness requires more of an involved explanation in practice, but is the converse of the theorem of soundness: Given the sense of a statement, there is a properly-construct-able sentence to which there exists a statement in the `first-order language` describing the sense of a statement which can then be compared to a `natural language`.

2.2 Context-free Grammars

The notion of grammar is fairly mathematical naturally, but perhaps that is the case as mathematics steals a lot from `natural language` in its derivation. In any case, it is hard not to make an analogy between the two when such notions as subject, object, predicate, copula, *etc.* are nearly identical without needing to compare them by complex methods.

Going into mathematical details, it can be said that there is a correspondence between mathematical statements and ordinary English sentences. Perhaps one point, however, that needs to be made more clear is the tricky ability that `natural language` has over mathematical statements in differentiating objects. English has two articles, *a/an* and *the*, and the usefulness of these articles is without limit; though probably not many people stop to think about the complexity that actually goes into using these articles appropriately, and how naturally people come to be able to use them so effectively. Furthermore, one could make conversation in a similar fashion about the demonstrative pronouns, *this*, *that*, *these*, and *those*, but these articles aid more to the complexity of `natural language` than the convenience of the well-spoken English user. Sentences of `natural language` are equally capable of existing without the addition of these pronouns, or other pronouns; but prior to these there must first be a notion of `context` by which pronouns may replace what they wish.

Articles affect nouns. A specific thing in the world can be defined, as *the noun* in question, or can remain to be indefinite, as *a noun* to be questioned. *A noun* can affect things in the universe by using *a verb*. *The noun* prior to *a verb* either acts in such a way that it affects *a different noun*, or it acts intransitively so as to only modify itself. *A preposition* may also more greatly specify the qualities of *a noun*.

These parts of speech, from a mathematical perspective, can be recursively linked together via a language construction called a `context-free grammar`. The following is an example of a `context-free grammar` for `natural language` construction:

$$
\begin{aligned}
\langle \text{SENTENCE} \rangle &\to \langle \text{NOUN-PHRASE} \rangle \langle \text{VERB-PHRASE} \rangle \\
\langle \text{NOUN-PHRASE} \rangle &\to \langle \text{CMPLX-NOUN} \rangle \mid \langle \text{CMPLX-NOUN} \rangle \langle \text{PREP-PHRASE} \rangle \\
\langle \text{VERB-PHRASE} \rangle &\to \langle \text{CMPLX-VERB} \rangle \mid \langle \text{CMPLX-VERB} \rangle \langle \text{PREP-PHRASE} \rangle \\
\langle \text{PREP-PHRASE} \rangle &\to \langle \text{PREP} \rangle \langle \text{CMPLX-NOUN} \rangle \\
\langle \text{CMPLX-NOUN} \rangle &\to \langle \text{ARTICLE} \rangle \langle \text{NOUN} \rangle \\
\langle \text{CMPLX-VERB} \rangle &\to \langle \text{VERB} \rangle \mid \langle \text{VERB} \rangle \langle \text{NOUN-PHRASE} \rangle \\
\langle \text{ARTICLE} \rangle &\to \text{a} \mid \text{the} \\
\langle \text{NOUN} \rangle &\to \text{boy} \mid \text{girl} \mid \text{flower} \\
\langle \text{VERB} \rangle &\to \text{touches} \mid \text{likes} \mid \text{sees} \\
\langle \text{PREP} \rangle &\to \text{with}
\end{aligned}
$$

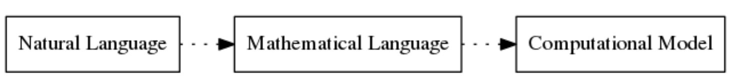
The following is an example derivation:

$$
\begin{aligned}
\langle \text{SENTENCE} \rangle &\Rightarrow \langle \text{NOUN-PHRASE} \rangle \langle \text{VERB-PHRASE} \rangle \\
&\Rightarrow \langle \text{CMPLX-NOUN} \rangle \langle \text{VERB-PHRASE} \rangle \\
&\Rightarrow \langle \text{ARTICLE} \rangle \langle \text{NOUN} \rangle \langle \text{VERB-PHRASE} \rangle \\
&\Rightarrow \text{a } \langle \text{NOUN} \rangle \langle \text{VERB-PHRASE} \rangle \\
&\Rightarrow \text{a boy } \langle \text{VERB-PHRASE} \rangle \\
&\Rightarrow \text{a boy } \langle \text{CMPLX-VERB} \rangle \\
&\Rightarrow \text{a boy } \langle \text{VERB} \rangle \\
&\Rightarrow \text{a boy sees}
\end{aligned}
$$

Since the rules of grammar make more sense to people naturally, the specifics of the `context-free grammar` will be kept to a minimum. We must trust ourselves to be capable of generating `natural language` statements, and leave the specifics of a `context-free grammar` to the later implementation details. There is a more important notion of `context generation` to come later that will clear any confusions on the subject. At the moment focus will be placed on the specifics of the `computational model` used to compute and verify `natural language` sentences.

2.3 Computing Languages

`Computational complexity` is measured by a machine's ability to understand languages. The model below describes the general idea behind the derivation process. In order that a `computational model` can exist, there must first be a `natural language` which can be reduced to a language generated by some `context-free grammar`. The statements generated by the `context-free grammar` can take the place of `sentence symbols` in the `first-order language`. This reduction process corresponds to a reduction of `natural language` to a `mathematical language`. From this point, since `mathematical languages` that exist are such that they abide by the soundness and completeness theorems, the problem can be further reduced to an algorithm which can be run on some kind of `computational model`. When a `computational model` emulates the desired `natural language` sentence, the statement is said to be verified.



Considering the `context-free grammar` in a prior section, the following statements may be generated:

P_1 A boy likes a girl with a flower.

P_2 The girl touches a flower.

P_3 The boy sees the girl with a flower.

$\mathbf{P_4}$ The boy likes the girl.

In ordinary English, we can make the statement: "If a boy likes a girl with a flower, and the girl touches a flower, and the boy sees the girl with a flower, then the boy likes the girl." The logical implication is that the three statements P_1, P_2, and P_3 are sufficient for the statement P_4 to be true, though perhaps the statement is a little contrived. An ordinary human would perhaps like to say the more natural phrase: "That boy must like that girl over there with the flower." The fact that the boy likes the girl is evident in the statement (or rather the speaker is testing the conviction of such a statement); just as well, the fact that there *is* a girl is evident (as per the usage of the demonstrative pronoun). However the idea that a boy *would* like a girl would be naturally formulated by `context`—the ordinary human speaker's most prominent point of contention is in the utilization of the word *must*, and it is probably that he/she is afterwards awaiting the agreement or rejection of the statement by the listener. Furthermore the idea that the girl *touching* a flower implies that the girl is *with* a flower would be formulated by `context` as well—imagine that the scene has many girls but there is only one flower, then the girl with the flower would be unique (which is sufficient condition for that statement to be true).

In any case, the reduction of the statement from the `natural language` to a `first-order language` is

$$(P_1 \wedge P_2 \wedge P_3) \vdash P_4.$$

The binary connective symbol, $\wedge$, corresponds to the English conjunctive, *and*. However this statement does not use the example `first-order language` previously described. This `first-order statement` is (by a bit of hand-waving) tautologically equivalent to

$$\neg(P_1 \rightarrow (P_2 \rightarrow \neg P_3)) \vdash P_4.$$

From this point, we can begin the derivation process by asserting that P_4 must hold. That is, let the statement "The boy likes the girl" hold. (Remember this is what the speaker would have been awaiting if there were `context` given to the statement. It was his/her conviction that the girl in question *must* like the boy in question.) We have derived from our `natural language` that $\neg(P_1 \rightarrow (P_2 \rightarrow \neg P_3)) \vdash P_4$ holds, and have already determined that P_4 holds, so that it must be that $\neg(P_1 \rightarrow (P_2 \rightarrow \neg P_3))$ holds. This statement, $\neg(P_1 \rightarrow (P_2 \rightarrow \neg P_3))$, holds whenever $P_1 \rightarrow (P_2 \rightarrow \neg P_3)$ does not. This negated statement does not hold if it is that P_1 holds but $(P_2 \rightarrow \neg P_3)$ does not. At this point, we can state that P_1 holds so it must be true that "A boy likes a girl with a flower." The statement $(P_2 \rightarrow \neg P_3)$ does not hold if it is that P_2 does hold, but the statement $\neg P_3$ does not, or rather that P_3 does hold. From this point we can state that P_2 holds so it is true that "The girl touches a flower," as well as that P_3 holds so it is true that "The boy sees the girl with the flower." In sum the statement reduced to a `first-order language` statement is valid, and would be accepted by another machine capable of generating the statement. Any other machine of similar `computational complexity` *is* capable of generating the statement; therefore this `natural language` sentence is verifiable.

3 The Human Automaton

Let's step away from the notion of language construction for a moment to talk about the humanity that our language construction is trying to simulate. The metaphysical question is whether or not there is some computational machine able to simulate the conversation of humans. The road taken thus far has talked about some of the building blocks that make `natural languages` resemble the chosen `mathematical languages` in some way, but has not yet come to touch on the ways in which humanity uses language.

Language is about communication. Language was developed as a method of delivering ideas between people or peoples. Language is the method by which we have come to comprehend intelligence. Were it not for language, then we would have no way to consider the complexity of the world around us; and it is with language that we take the simple things around us and try to compute more complex ideas, which in turn generates more complex language statements. The `human automaton` uses a similar concept. The `human automaton` starts from some `context` generated from human understanding, and attempts to create new idea from it using logic. Listening to another person speak is really just verification of `natural language statements`. A typical conversation will have a speaker explaining some situation or some new idea, a new `context`, until the listener of the conversation interrupts him; either to verify comprehension of the idea or situation, or to ask that a certain point be elaborated on or reiterated so that the explanation can continue again until the new `context` is understood. At the end of the conversation the listener will have generated new ideas as a result of understanding this new `context`, and he/she can now be a speaker for his/her own, now greater, `context`.

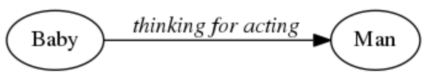
At the moment, there is not a model for generating this kind of `context`, and so there is not anything in place allowing a possible machine to generate conversation. That is, even when using proper, grammatically-correct language, there cannot be any meaning behind statements generated by a `context-free grammar`. The `context-free grammar` can generate `natural language` statements, these statements are able to be composed within a `first-order language`, and there is at least some meaning behind the words in that the words act as a part of speech valid according to the sentence construction; however there is no meaning behind the statements, as there is no `context` given to them—they are, at this point, randomly generated statements placed arbitrarily, as `sentence symbols`, into a `first-order language`. The unique idea behind the `human automaton` is that there is no beginning `context` to which new `context` is generated around. A human is born one day, knows nothing, and then someday magically comes to understand the `context` of the world around; he/she somehow emerges into the world of conscience conversation, then begins constructing verifiable, grammatically-correct, `natural language` sentences.

3.1 The Life of the Human

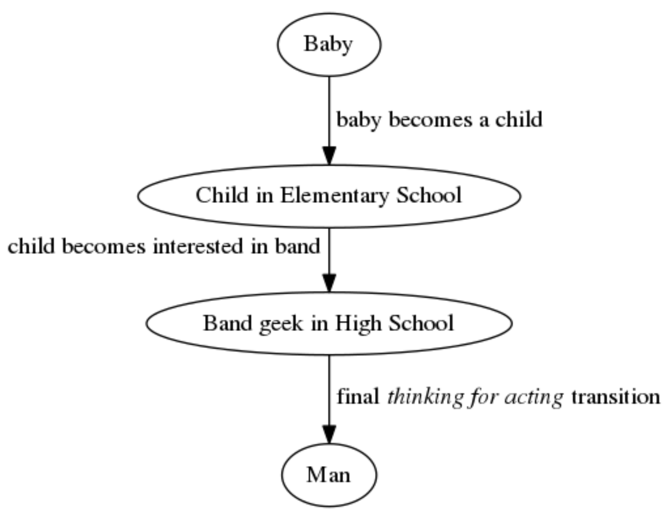
The life of the human starts at birth. Though let this be a disambiguation from the life of the man, as the life of the man begins when the human has fully learned *thinking for acting*. Consider this an important distinction, as the young human, the baby, cannot *think* fully yet and so cannot fully *act* either. The baby does not learn to *think* and *act* fully until he/she becomes a man, or rather he/she fully becomes a man/woman the day that he/she learns to *think* and *act*. The following is a `transition diagram` emphasizing this process[1]:

Figure 1: generalized human upbringing

In same fashion, the `transition diagram` will act as the `context` by which the `human automaton` operates. The idea of this operational process might be a little more involved in considering the complexity of the *thinking for acting* `transition`. That is, the `transition` can be expanded. Any `transition` on the `transition diagram` has the capacity to be expanded.



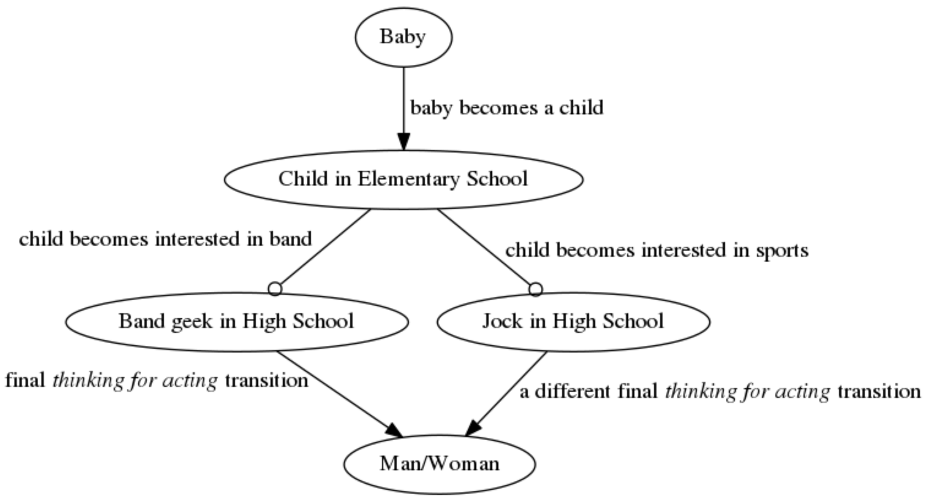
Since people are different, any single person will take their own *thinking for acting* `transition` differently. That is to say, one may take their *thinking for acting* `transition` as a series of important `transitions` in their life in such a way that is unique to themselves. Obviously, the baby must first become a child before he/she becomes a teenager, and, maybe, the child will become interested in high school marching or concert band as a teenager before he/she finally becomes a man. Following this example is another `transition diagram`:



This `transition diagram` is `singular`. It represents the life of only one individual. There is only one clear path to describe the series of `transitions`, and while this `transition diagram` may not be capable of describing every single person's identity throughout his/her life, it is certainly capable of describing one person's. Therefore the description of this

[1]where the square represents an `event` in time, the circles represent the `states` of the person, and the arrows are the `transitions` between the objects—the dotted arrow is a `resultant transition`, indicating that an event results in a state of being whereas the solid arrow is a `traditional transition`—or just a `transition`—indicating the change from one `state` to another `state`.

`transition diagram` is limited in some way. Ideally this particular person who ascribes his/her life to this `transition diagram` would like to have for himself more options, and certainly there could be a plethora of these options (and furthermore any of these options could be expanded, as any `transition` on the `transition diagram` has the capacity to be expanded). Thus where this `transition diagram` is `singular` and fairly simple, there is a `transition diagram` where the `transitions` are `variable`. Continuing the example is a `variable`[2], `transition diagram` describing the life of the child as the superposition of his/her life being a high school football quarterback with that of his/her life being a high school band geek.



3.2 Comparison to Mathematical Models

To compare the similarities of the human `transition diagram`, a similar `transition diagram` will be fixed on the basic principle of a `division algorithm`. The discussion will refer to this algorithm using the form

$$N \setminus D = (Q, R),$$

where:

- $N :=$ `Numerator (dividend)`
- $D :=$ `Denominator (divisor)`

is the input, and

- $Q :=$ `Quotient`
- $R :=$ `Remainder`

[2]The open dot at the end of a transition refers to the transition being `variable`. In some cases it may also be referred to as `non-deterministic`, as a `singular` decision cannot be `deterministic` given `variable transitions`.

is the output.

Just as the lives of men unfortunately come to an end, so too does the life of this `division algorithm`. It is thus called a `terminating algorithm`. As an example on given inputs $N = 96$ and $D = 7$, the output is produced as $Q = 13$ and $R = 5$.

At this point, `transition diagram` does not show the `division algorithm` to be very algorithmic. The `transition diagram` only includes one `transition` when usually an algorithm involves a series of steps. Thus in the same way that the life of the man was expanded earlier, the steps of the algorithm can be expanded. The `division algorithm` as above can be defined in an innumerable number of ways by being tricky. To keep things simple, however, the definition of the `division algorithm` here will use the method presented in Euclid's *Elements*, which finds the remainder upon division using only subtractions and comparisons. At each transition, the step number is incremented and the value of the `denominator` is subtracted from the `numerator`. When the `denominator` can no longer be negated from the `numerator` without producing a negative number, the algorithm terminates. The `singular transition diagram` of this algorithm follows from the full division process figure.

Returning to the human automaton analogy where the `transitions` of the man could be `variable`, the `transitions` of the division can be made `variable` also. One who is well-versed in the ways of dividing numbers could consider first doing 10 steps of the `division algorithm` in one fell swoop, moving from the 0^{th} step to an alternative 10^{th} step, at which point one could notice that moving 3 steps forward—to that of the alternative 13^{th} step—would be the maximum number of additional steps possible to take. Another person might do the normal algorithm until he/she reaches the 3^{rd} step of the algorithm, at which point the `numerator` sits at 75, a number reducible to 10 additional steps of the `division algorithm` all at once, making it obvious to move to the alternative 13^{th} step of the algorithm by the reason that the `numerator` 75 is at most 10 of the `denominator` 7. This is detailed in the full division process with variability figure.

As a remark, this sort of diagram is analogous to how the neurons of a human brain work. In this example the analogy should be particularly obvious. A person who is very well-versed in the ways of division would not have a hard time at all moving through this diagram to produce the answer. That is, the network of neurons in his/her head which facilitate the production of the solution is *myelinated*. Some of the `variable transitions` of this diagram which facilitate the retrieval of the final solution or of the neural network in the brain of the well-versed human divider are correspondingly properly *myelinated*, and therefore the decision-making process which results in the movement through either is very rapid. Markov chains are considered the probabilistic analogous to automatons, and could facilitate in making the analogy between the `transition diagram` and the neural networks of the human brain.

3.3 AI-hard

Constructing a kind of `human automaton` for the purpose of understanding natural language is the `Natural Language Understanding` (NLU) problem in the field of artificial intelli-

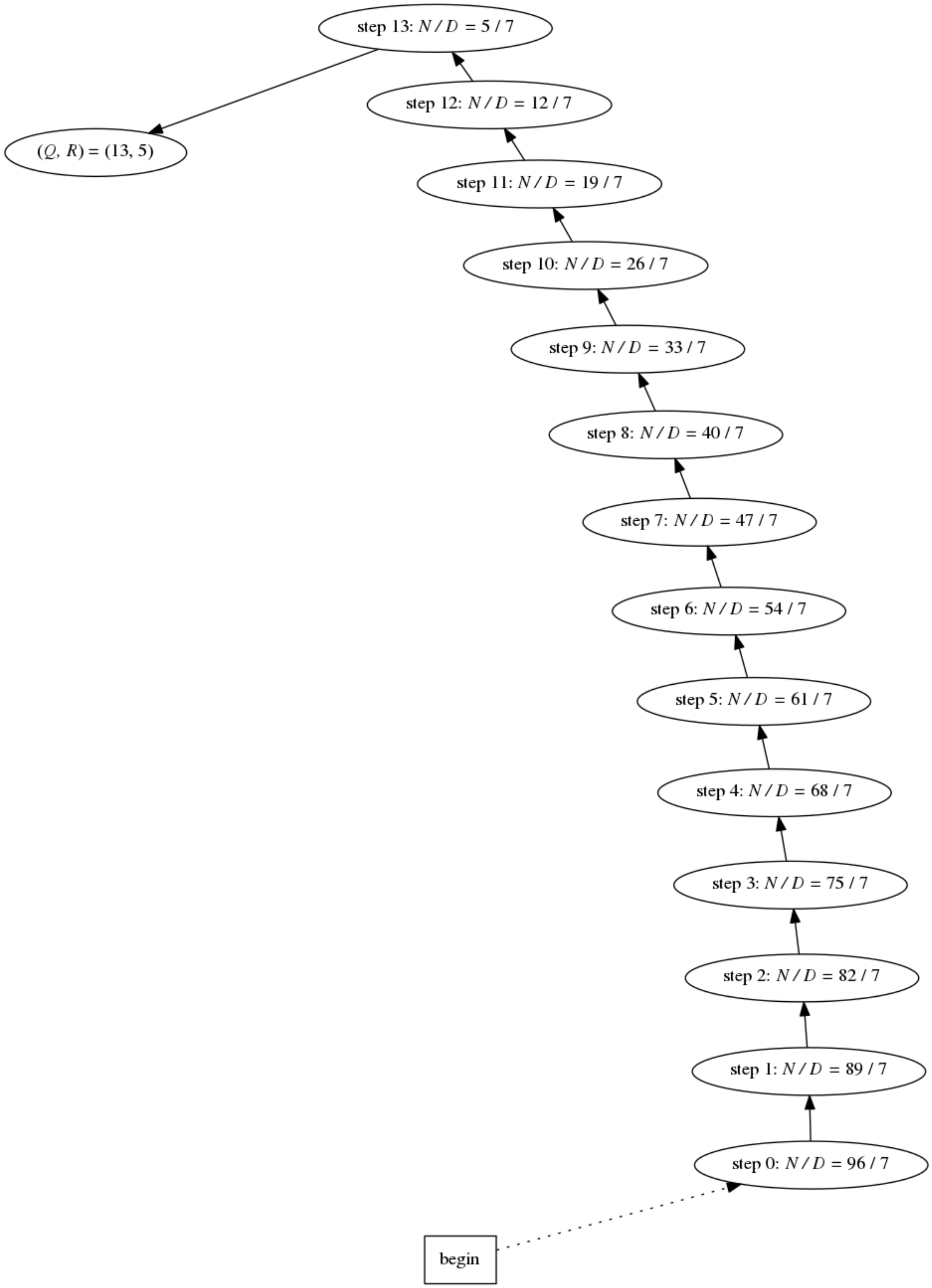


Figure 3: full division process

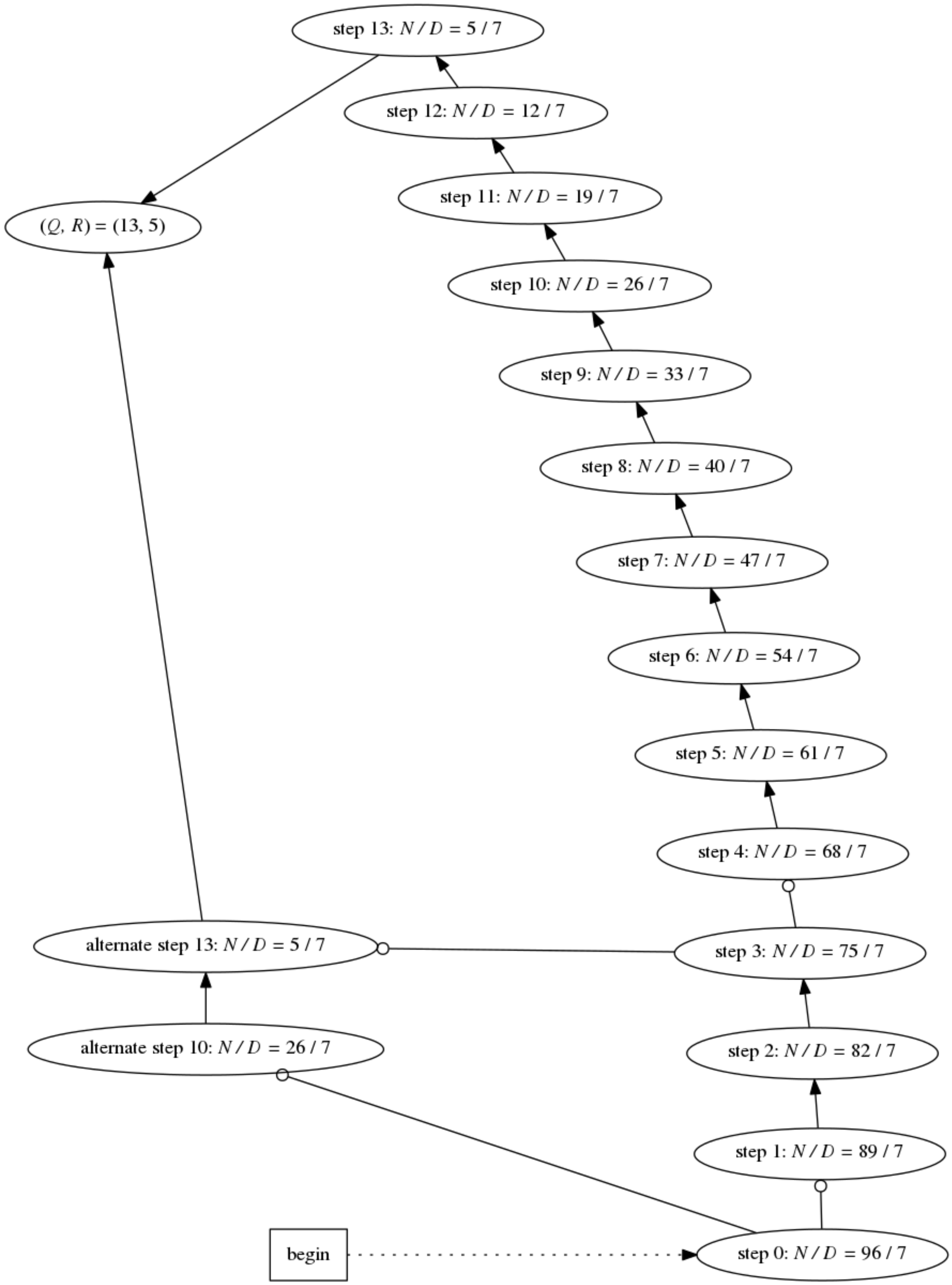


Figure 4: full division process with variability

gence. It is an AI-hard problem, and it is said that *simulating* a machine which understands natural language completely corresponds to *duplicating* human intelligence.

The study of language is typically divided into three separate categories: syntax, semantics, and pragmatics. Pragmatics is, essentially, how syntax and semantics operate together to formulate language, so pragmatics in the study of language here will be ignored. The point of this paper ultimately answers the question supposed by pragmatics, so specifically concerning ourselves with pragmatics is metalogically useless. The other two categories, however, provide the very questions that need to be answered. Since computers are syntactically bound, the question posed by syntax becomes a question of logistics; that is, how the structure of the computational model should be. This is a question best left to the implementation details, which is better considered later. The other category, semantics, poses a tricky problem inherent to the study of AI.

Thus far, there has been provided a simple construction of the syntactical elements necessary for computers to develop its understanding of language; syntactical understanding of phrases is naturally very computational, because computation itself relies specifically on the elements of syntax. The bigger problem in the field lies in a computer's inability to provide semantic information to the understanding of language. When a human hears or reads a natural language statement, his/her understanding of the statement is facilitated by a robust understanding of the meaning of the words and phrases inherent to the statement. A computer looking only into the syntactical derivation of a statement still has no conscience understanding of the words or phrases; and therefore could not possibly understand the statement. It makes sense then that providing a method by which computers could understand the words, phrases, and statements therefore provides the computer with intelligence; possibly even implying that the computer is being provided with consciousness.

In any case, to begin to at least simulate the way in which a human might think about the world, let us continue to consider how the notion of the `transition diagram` might be able to operate as a `context`, beginning this intelligence-simulating path.

4 Context Generation

4.1 Rules of Inference

The ideal method to derive the logic from a given context lies in a deductive reasoning system. That is, given some antecedent statement A, one wants to infer B by some implication $A \to B$. Since A was given, one can write $A, A \to B$. Then, by *modus ponens*, $A, A \to B \vdash B$. Unfortunately, using deductive reasoning in a human environment is not always possible. Human thought is causal. A human takes some sensory stimulus from one of his/her senses which in turn inspires some thought relevant to his/her context. The conclusion is some epistemically objective, observer independent observation holding true, and the thoughts are what is inferred about the conclusion; the observer wants to believe that his/her thoughts, the inferred antecedent logic, led to the conclusion witnessed. Thus in a human environment one must use an abductive reasoning system. A stimulus causes one to derive an implication from his/her context which led to the conclusion, wherein the statements of the context function as the antecedent which hypothetically imply the conclusion. That is, one would like to say that $A, B \vdash A \rightarrow B$. However, given the definition of $\rightarrow$, this goes against the definition of the `first-order language`.

From a computational standpoint, it may be hard to determine which antecedent context possibly leads to the hypothetical implication as well. A human has a limitless reference bank to which his thoughts may traverse. Given a sequence of contextual statements $A_1, A_2, ...A_n$ how could one determine which statement, or set of statements $X \subseteq \{A_1, A_2, ..., A_n\}$ actually led to the hypothetical implication of the statement B? Written

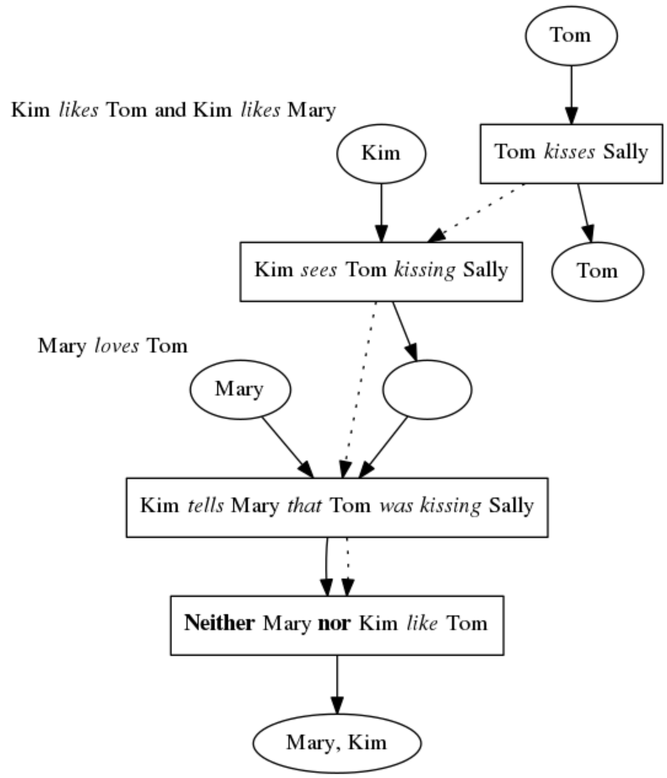
$$\left(\bigwedge_{x \in X} x, B\right) \vdash \left(\left(\bigwedge_{x \in X} x\right) \rightarrow B\right).$$

4.2 Language Generation

Let there be a `context-free grammar` consisting of

- verbs *see*, *kiss*, *tell*, *like*, and *love*;
- proper nouns Kim, Tom, Mary, and Sally;
- subordinating conjunction *that*; and
- unary logical operator **not** (mask-able as **neither**… **nor**, or **no** where convenient).

The following is a `transition diagram` implementing statements possibly generated by the `context-free grammar`. Note in this `transition diagram` how `transitions` and `events` are treated similarly. Note also that `events` in this diagram are `reactive`, meaning one `event` `deterministically` causes the next `event`; so the `resultant transition` points to another `event`. This `transition diagram` possibly produces something in the `natural language` English as "Tom kisses Sally, but Kim sees Tom kissing Sally. Later Kim tells Mary that she saw Tom kissing Sally. Now neither Mary nor Kim like Tom."

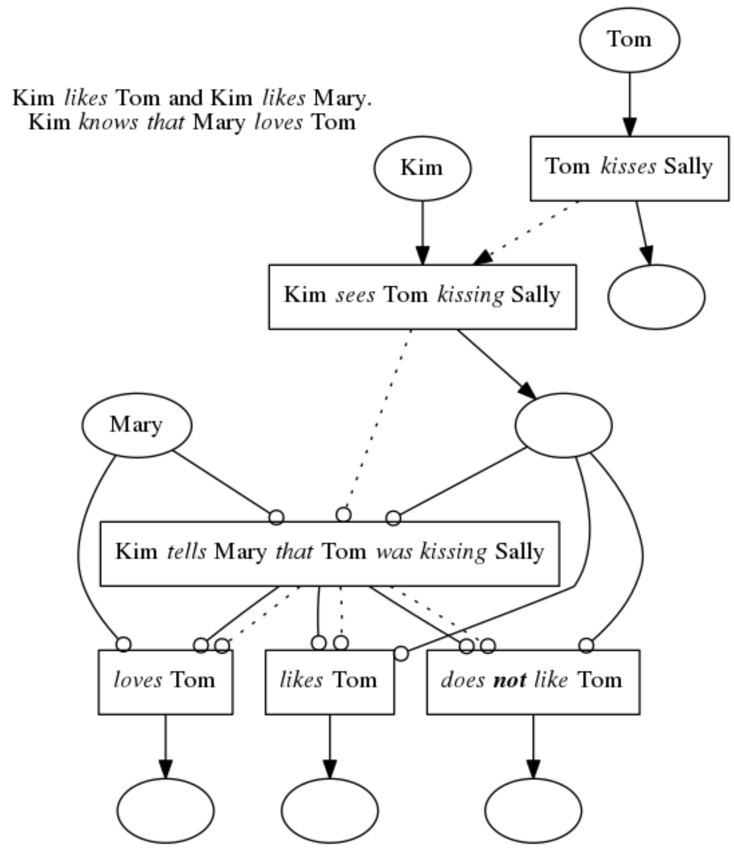


This `transition diagram` represents a `singular context` by which some person, or some automaton, interprets a series of `events`. There are, however, other possibilities by which this series of `events` could have unfolded themselves; considerations of these other possibilities generate `non-deterministic`, `variable transitions` in a more complex `transition diagram`.

The following `variable transition diagram` takes into consideration the possibility that Kim *does **not** tell* Mary *that* Tom *was kissing* Sally. This is represented by the changing of the `singular transition` previously transitioning Kim into *telling* Mary *that* Tom *was kissing* Sally into `variable transitions`: one the same as before, the other two such that Kim does not inspire any `events`, but does transition her into possibly feeling differently about Tom (either she continues to *like* Tom, or she finds that the event transitions her into ***not*** *liking* Tom anymore). When Kim does not transition herself into the *telling* Mary *that* Tom *was kissing* Sally, however, the `event` inspired by Kim *seeing* Tom *kissing* Sally fails to transition Mary into a next appropriate `state`. Thus there needs to be more `variable transitions` made so that Mary can find herself somewhere later in the `transition diagram`. Since there was no new `event` to inspire Mary to be anywhere, the new event by which she transitions herself is `idempotent`, or does nothing, but satisfies the requirement that she finds herself in an acceptable state later in the `transition diagram`. That is, Mary *loves* Tom in the beginning of the diagram, and Mary *loves* Tom at the end of the diagram.

There are also new `variable transitions` and new `events` made to detail the different possible inspirations that the `event` might have had on the two people in the Kim *telling* Mary that Tom *was kissing* Sally `event`. Perhaps Mary continues to *love* Tom even after the conversation, or perhaps she is reduced to only *liking* Tom, or perhaps she decides that she ***no*** *longer likes* Tom. Kim would likely transition into either liking Tom or ***not*** *liking* Tom, but the possibility that she somehow *loves* Tom after the conversation is not ruled out as a possibility either.

Finally, the `events` previously described to be `reactive events` that were previously `deterministic`, but now made to be `non-deterministic`, are called `promised events` (that is, the `event` may not happen because events can be canceled in the same way that promises can be broken).



4.3 Logic Generation

Let's take a step back to the very beginning. The `natural language` sentence constructed from the `singular context` was "Tom kisses Sally, but Kim sees Tom kissing Sally. Later Kim tells Mary that she saw Tom kissing Sally. Now neither Mary nor Kim like Tom." Let's consider writing this sentence as statements generate-able by some `context-free grammar`:

$\mathbf{P_1}$ Tom *kisses* Sally,

$\mathbf{P_2}$ Kim *sees* Tom *kissing* Sally,

$\mathbf{P_3}$ Kim *tells* Mary *that* Tom *was kissing* Sally,

$\mathbf{Q_1}$ Kim *does* ***not*** *like* Tom, and

$\mathbf{Q_2}$ Mary *does* ***not*** *like* Tom.

The `natural language` sentence "Tom kisses Sally, but Kim sees Tom kissing Sally" can be formulated as a `first-order logic` statement $P_1 \wedge P_2$. This statement logical implies the following `event` which corresponds to the `natural language` sentence "Kim tells Mary that Tom was kissing Sally." So, $(P_1 \wedge P_2) \vdash P_3$. The ensemble of each of these `events`, and their corresponding statements, results in the logical implication of the first-order logic statement $Q_1 \wedge Q_2$. In sum,

$$((P_1 \wedge P_2) \vdash P_3) \vdash (Q_1 \wedge Q_2).$$

Written in our example `first-order language` (again by a bit of hand-waving), this is tautologically equivalent to

$$(\neg(P_1 \to \neg P_2) \vdash P_3) \vdash \neg(Q_1 \to \neg Q_2).$$

As before, let's start by assuming that our consequent was true, *i.e.* that statement $\neg(Q_1 \to \neg Q_2)$ holds. This statement holds when the statement $Q_1 \to \neg Q_2$ does not hold, or when Q_1 holds but $\neg Q_2$ does not. Q_1 holding means that "Kim *does **not** like* Tom" is a true statement. $\neg Q_2$ does not hold when Q_2 does hold, so "Mary *does **not** like* Tom" is a true statement. The antecedent of the logical implication must hold because $(\neg(P_1 \to \neg P_2) \vdash P_3)$ logically implies the consequent, and the consequent holds. Thus $(\neg(P_1 \to \neg P_2) \vdash P_3)$ means that $\neg(P_1 \to \neg P_2)$ holds so that P_3 holds. P_3 holding corresponds to the statement "Kim *tells* Mary *that* Tom *was kissing* Sally" being true. (Note the correspondence of this `first-order logic` statement P_3 to that of the two `transition diagrams` from before. In the `singular context` it is asserted, via the usage of the $\vdash$ symbol, that the `event` corresponding to the statement P_3 actually happened because the two events corresponding to P_1 and P_2 actually happened. The previous two `events` happening, corresponding to P_1 and P_2 holding, logically implies the happening of the `event` in correspondence with P_3.) Since $\neg(P_1 \to \neg P_2)$ holds, it must be that $P_1 \to \neg P_2$ does not hold. $P_1 \to \neg P_2$ does not hold when P_1 holds, but $\neg P_2$ does not, or P_2 does hold. Thus, it is that the sentence "Tom *kisses* Sally" is true, and the sentence "Kim *sees* Tom *kissing* Sally" is true.

As before when constructing the `variable transition diagram`, the same consideration that the `event` corresponding to statement P_3 not happening will be made. This changing of the diagram corresponds to a changing of the `first-order logic`. The first-order statement

$$(\neg(P_1 \to \neg P_2) \vdash P_3) \vdash \neg(Q_1 \to \neg Q_2).$$

becomes

$$(\neg(P_1 \to \neg P_2) \to P_3) \vdash \neg(Q_1 \to \neg Q_2).$$

As such, no longer does the statement $\neg(P_1 \to \neg P_2)$ logically imply P_3, and it is no longer possible to deduce absolute truth from the statement. $(\neg(P_1 \to \neg P_2) \to P_3)$ does not hold when the statement $\neg(P_1 \to \neg P_2)$ holds, and P_3 does not. Since we are making the consideration that "Kim *does **not** tell* Mary *that* Tom *was kissing* Sally," then $\neg P_3$ holds and therefore P_3 does not. Therefore the statement $(\neg(P_1 \to \neg P_2) \to P_3)$ does not hold. Furthermore, it does not hold that

$$(\neg(P_1 \to \neg P_2) \to P_3) \vdash \neg(Q_1 \to \neg Q_2),$$

but it can be reconstructed as

$$(\neg(P_1 \to \neg P_2) \to P_3) \to \neg(Q_1 \to \neg Q_2).$$

This corresponds to the considerations that were made in the reconstruction of the `singular transition diagram` to the `variable transition diagram`. Since the statement $(\neg(P_1 \to \neg P_2) \to P_3)$ does not hold, then the consequent $\neg(Q_1 \to \neg Q_2)$ could either hold or not hold for the statement $(\neg(P_1 \to \neg P_2) \to P_3) \to \neg(Q_1 \to \neg Q_2)$ to hold. This means that either Q_1 or $\neg Q_1$ could hold, and either Q_2 or $\neg Q_2$ could hold. Q_1 holding means that "Kim *does **not** like* Tom," while $\neg Q_1$ holding means that "Kim *does like* Tom." Q_2 holding means that "Mary *does **not** like* Tom" (or possibly "Mary *loves* Tom"), while $\neg Q_2$ means that "Mary *does like* Tom."

Possible new `natural language` sentences may be derived and verified as follows:

- "Tom kisses Sally, but Kim sees Tom kissing Sally. Later Kim doesn't tell Mary that she saw Tom kissing Sally. Mary still loves Tom, while Kim has come to not like Tom."
- "Tom kisses Sally, but Kim sees Tom kissing Sally. Later Kim doesn't tell Mary that she saw Tom kissing Sally. Mary still loves Tom, and Kim still likes Tom even though she knows Tom's little secret."
- "Tom kisses Sally, but Kim sees Tom kissing Sally. Later Kim doesn't tell Mary that she saw Tom kissing Sally. Mary still loves Tom. Kim's knowing the little secret about Tom and Sally has come to make her realize that she herself loves Tom."

4.4 Context Generation

The reorganizing and re-contextualizing of the `language generation` and `logic generation` processes represents the idea of `context reconciliation`. In other words, by taking a specific `singular context`, reconsidering an `event` of the context and its resulting reconstruction of `events`, a `variable context` can be created; there is a `first-order language` analogous to this reconsideration of events and therefore a `computational model` capable of devising new, meaningful `natural language` sentences given by the vocabulary inherent to the `context-free grammar`.

This process, however, relies upon a need to have `contexts` already build into the `automaton`. When trying to understand logic of a variable context, the `automaton` looks upon `contexts` which it previously has verified.

The process is as follows, where $\langle C \rangle$ is some new context taken from human life, or generated by another automaton:

Def 4.1. Human Automaton Context Generator. $HACG =$ "On input $\langle C \rangle$:

1. Generate the `first-order logic` statement from the context $\langle C \rangle$.
2. Recursively split the left- and right-hand sides of the highest-level binary connective symbol, deducing the logical possibilities of the broken up `natural language` statements where atomic sentence symbols are hit.

3. Draw a new `non-deterministic transition diagram` from the new `first-order language` statement possibilities.

4. Draw new `singular contexts` from `non-deterministic transition diagram` by taking all possible `singular transitions` and generate `natural language` statements.

5. Verify conclusions from `step 4`'s `natural language statements` in closely-related, previously-verified `contexts` and write newly-constructed contexts to memory.

6. When every new `context` is made and verified, halt and *accept*. If no new contexts were generated, then throw away the `context` $\langle C \rangle$ and *reject*."

5 Bibliography

- Boroditsky, Lera; Schmidt, Lauren; Phillips, Webb, "Sex, syntax, and semantics" (PDF), Language in Mind: Advances in the Study of Language and Thought, pp. 61–79

- Carroll, Lewis (1896). Symbolic Logic. London, New York, Macmillan.

- Euclid, ., Heath, T. L., & Densmore, D. (2002). Euclid's Elements: All thirteen books complete in one volume : the Thomas L. Heath translation. Santa Fe, N.M: Green Lion Press.

- Koutsofios, E., & North, S. (1996). Drawing graphs with dot. dot User's Manual, AT&T Bell Laboratories, Murray Hill, NJ.

- Ovchinnikova, E. (2012). Integration of World Knowledge for Natural Language Understanding Volume 3 of the series Atlantis Thinking Machines pp 15-37. Atlantis Press.

- Pustejovsky J., et. al. (2006) TimeML Annotation Guidelines Version 1.2.1.

- Sipser, M. (2006). Introduction to the theory of computation. Boston: Thomson Course Technology.

- Spinoza, B. ., Hampshire, S., & Curley, E. M. (1996). Ethics. London: Penguin Books.