(a) Quantifiers always introduce variables to their left and parentheses are used normally.

(b) No quantifier can be introduced without a variable and n-ary relations from the model vocabulary must always include n variables.

(c) Many beginning students make this sort of error when trying to express a logical sentence which forbids *nt* sequences. This expression breaks the same rules as the one before it.

We conclude this section by providing an example of a logical sentence defining a constraint which bans voiceless obstruents after nasals. This is constraint in the literature is often abbreviated *NT. Since the model signature does not include relations for concepts like nasals and voiceless consonants, we first define predicates for these notions. We assume the alphabet is limited to the following IPA symbols: *a,b,d,e,g,i,k,l,m,n,o,p,r,s,t,u,z*.

Example 4 (The constraint *NT defined under the FO with successor model.)**.**

$$\mathsf{nasal}(x) \stackrel{\text{def}}{=} \mathtt{n}(x) \vee \mathtt{m}(x) \tag{2.1}$$

$$\mathsf{voiceless}(x) \stackrel{\text{def}}{=} \mathtt{p}(x) \vee \mathtt{t}(x) \vee \mathtt{k}(x) \vee \mathtt{s}(x) \tag{2.2}$$

$$\text{*}\mathsf{NT} \stackrel{\text{def}}{=} \neg\exists x, y(x \lhd y \wedge \mathsf{nasal}(x) \wedge \mathsf{voiceless}(y)) \tag{2.3}$$

It is easy to see that models of words like *tent* and *lampoon* do not satisfy *NT but models of words like *ten* and *moon* do. For example, in the model of *tent*, the expression $\exists x, y(x \lhd y \wedge \mathsf{nasal}(x) \wedge \mathsf{voiceless}(y))$ is true when $x = 3$ and $y = 4$. Hence, *NT evaluates to false. On the other hand, in the model of the word *moon*, every value assigned x and y results in the sentence $\exists x, y(x \lhd y \wedge \mathsf{nasal}(x) \wedge \mathsf{voiceless}(y))$ evaluating to false. Hence the sentence *NT evaluates to true and so $\mathcal{M}_{moon} \models$ *NT.

This section has presented the first CDL: FO with successor. The FO with successor model has been studied carefully and it is known precisely what kinds of constraints can and cannot be expressed with this CDL (Thomas, 1982), as will be discussed below.

2.5 Feature-based Word Models

One way in which the successor model above is strange from a phonological perspective is its absence of phonological features. The properties associated

with the elements of the domain are whole segments. However, nothing in model theory itself prohibits domain elements from having more than one property. It is a consequence of the construction in Table 2.8 that each domain element will satisfy exactly one of the unary relations $\mathtt{a}$, no more and no less. We can formalize this statement of the successor model in Remark 1 as follows.

Remark 1 (The successor model entails disjoint unary relations)**.** For all successor models $\mathcal{M} = \langle D \mid (\mathtt{a})_{a\in\Sigma}, \triangleleft \rangle$, and for all $\mathtt{a},\mathtt{b} \in (\mathtt{a})_{a\in\Sigma}$, it is the case that $\mathtt{a} \cap \mathtt{b} = \varnothing$.

Therefore it is possible to design different models of words, where the unary relations do not represent segments like *a, b*, or *n* but phonological features such as *vocalic, labial*, or *nasal*. Crucially, in these models would not entail disjoint unary relations: a domain element could be both *voiced* and *labial* for instance.

In this part of the chapter, we give one example of such a model. There are many others, as many as there are theories of phonological features. The model we give here is primarily for pedagogical reasons; we are not stating particular beliefs or arguments regarding the nature of feature systems. We are only choosing a simple system that illustrates some key points.

We set up a feature system with **privative** features for the simple alphabet Σ discussed earlier *a,b,d,e,g,i,k,l,m,n,o,p,r,s,t,u,z*. The use of privative features contrasts with the typical assumption in phonological theory that features are binary. We choose not to pick a minimal nor maximal set of features for distinguishing this set. Instead we choose somewhat arbitrarily a middle ground based on standard descriptive phonetic terms used for describing the manner, place and laryngeal qualities in articulating sounds. We call this model the "successor model with features." Its signature is shown below.

$$\langle D \mid \mathtt{vocalic}, \mathtt{low}, \mathtt{high}, \mathtt{front}, \mathtt{stop}, \mathtt{fricative}, \mathtt{nasal}, \mathtt{lateral} \\ \mathtt{rhotic}, \mathtt{voiced}, \mathtt{voiceless}, \mathtt{labial}, \mathtt{coronal}, \mathtt{dorsal}, \triangleleft \rangle \quad (2.4)$$

Table 2.4 shows how to construct a successor model with features for any string in Σ^*. Again this model ensures that distinct strings from Σ^* have different models and that every string has some model.

Figure 2.2 shows the successor model with features of the word *tent*.

D	$\stackrel{\text{def}}{=}$	$\{1, 2, \ldots n\}$
`vocalic`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i \in \{a, e, i, o, u\}\}$
`low`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i = a\}$
`high`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i \in \{i, u\}\}$
`front`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i \in \{e, i\}\}$
`stop`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i \in \{b, d, g, k, p, t\}\}$
`fricative`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i \in \{s, z\}\}$
`nasal`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i \in \{m, n\}\}$
`lateral`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i = l\}$
`rhotic`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i = r\}$
`voiced`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i \in \{b, d, g, z\}\}$
`voiceless`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i \in \{k, p, s, t\}\}$
`labial`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i \in \{b, p, m\}\}$
`coronal`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i \in \{d, s, t, z\}\}$
`dorsal`	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i \in \{d, g, k\}\}$
$\lhd$	$\stackrel{\text{def}}{=}$	$\{(i, i+1) \mid 1 \leq i < n\}$

Table 2.4: Creating a successor model with features for any word $w = a_1 a_2 \ldots a_n$.

The successor model with features contrasts sharply with the successor model with features in an important way. To see how, first consider the constraint *NT. Under the successor model with features, this constraint would be defined as in Example 2.5

Example 5 (The constraint *NT defined under the FO with successor model with features.)**.**

$$\textsf{*NT} \stackrel{\text{def}}{=} \neg\exists x, y(x \lhd y \wedge \texttt{nasal}(x) \wedge \texttt{voiceless}(y)) \tag{2.5}$$

This looks similar to the definition of *NT under the successor model (Example 2.1), but there is a critical difference. The predicates above in Example 2.5 are *atomic* formula and not user-defined predicates as they are in Example 2.1.

$$
\begin{aligned}
\mathcal{M}_{tent} = \\
\Big\langle \{1,2,3,4\} \mid \texttt{vocalic} &= \{2\}, \\
\texttt{front} &= \{2\}, \\
\texttt{stop} &= \{1,4\}, \\
\texttt{nasal} &= \{3\}, \\
\texttt{voiceless} &= \{1,4\}, \\
\texttt{coronal} &= \{1,3,4\} \\
\triangleleft &= \{(1,2),(2,3),(3,4)\} \Big\rangle
\end{aligned}
$$

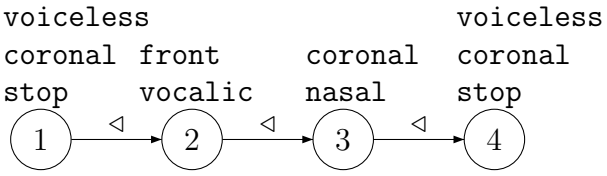

Figure 2.2: At left, the successor model with features of the word *tent*. Unary relations which equal the empty set are omitted for readability. At right, a graphical diagram of this model.

This is an important ontological difference between these two models. In the successor model with features there is no primitive representational concept that corresponds to a sound segment like t like there is in the successor model without features. Conversely, in the successor model without features there is no primitive representational concept that corresponds to a phonological like *voiceless* like there is in the successor model with features. In the successor model with features we can write user-defined predicates that define properties of domain elements that we can interpret to mean "being t".

$$\textsf{is\_t}(x) \stackrel{\text{def}}{=} \texttt{stop}(x) \wedge \texttt{coronal}(x) \wedge \texttt{voiceless}(x) \tag{2.6}$$

Other sound segments would be defined similarly.

One way to put this difference is that in the successor model with features one can immediately determine whether a domain element is voiced or not, but in the successor model without features one cannot immediately determine this fact. Instead one can deduce it by checking the appropriate user-defined predicate. Likewise, in the successor model with features one cannot immediately determine whether a domain element is t or not. With the featural representations, such a fact must be deduced with a user-defined predicate like the one above.

Also, the fact that such user-defined predicates exist should not be taken for granted. They exist here because the only logical system discussed so

far is FO. With FO logic, it is possible to define a predicate for any subset of the alphabet Σ for both successor models with and without features. If the logical system was restricted in some further way then some user-defined predicates may not be possible to define. For example, if the logical system only permitted conjunction and no other Boolean connective then it would not be possible to define a predicate for voiceless stops in the successor model without features. This interplay between representations and logical power with respect to expressivity is an important theme of this chapter. It will be discussed at length with respect to the successor relation, and we will return to it in the context of features when restricted logics are introduced towards the end of the chapter.

As a consequence of FO logic then, any constraint definable with one of the representations discussed so far is definable in the other. This leads to the conclusion that there are no typological distinctions between the FO with successor theory and the FO with successor with features theory. Both admit exactly the same class of constraints.

However, while the two models do not make different typological predictions, they do make different psychological ones. In regard to phonological theory, the signature of the model is an ontological commitment to the psychological reality of the model vocabulary. Taken seriously, the successor model with features says that the mental representations of words carries only the information shown in Figure 2.2. Thus, taken seriously, the successor model with features says that the segments in the word *tent* are not perceived as such but are instead perceived in terms of their features. Clever psycholinguistic experiments might be able to bring evidence to bear on which model more accurately resembles them actual mental representations of words.

2.6 Monadic Second-Order Logic

This section introduces Monadic Second-Order (MSO) logic. This logic is strictly *more expressive* than FO logic. We motivate the discussion of MSO logic from a linguistic perspective by showing that FO with successor, both with and without features, is not sufficient to account for long-distance phonotactic constraints.

What are long-distance phonotactic constraints? Odden (1994) draws attention to an unbounded nasal assimilation in Kikongo whereby underlying

/ku-kinis-il-a/ becomes [kukinisina] 'to make dance for.' From one perspective, this assimilation could be said to be driven by a phonotactic constraint that forbids laterals from occuring after nasals. Similar long-distance constraints have been posited for a variety of long-distance assimilation and dissimilation processes (Hansson, 2010).

We first show that the phonotactic constraint which bans laterals from occuring *anywhere* after nasals cannot be expressed in the FO with successor model. As we hope to make clear, the problem is that the notion of *precedence* is not FO-definable from successor. To illustrate, in Kikongo, [kukinisila] is an ill-formed string. The nasal has only one successor [i], but [n] *precedes* many segments including the second and third [i]s and the [s,l] and [a]. It is the fact that [n] precedes [l] which makes [kukinisila] ill-formed according to the phonotactic constraint which bans laterals from occuring *anywhere* after nasals. We refer to this constraint as *N..L.

Constraint *N..L is not FO definable with successor. To prove this we use an abstract characterization of the constraints definable with FO and successor due to Thomas (1982) and reviewed in Rogers and Pullum (2011).

Theorem 1 (Characterization of FO-definable constraints with successor)**.** *For every number t and every number n let the t-number of n equal n if $n < t$ otherwise let it be t. A constraint is FO-definable with successor if and only if there are two numbers k and t such that for any two strings w and v, and for all substrings x of length k, if the t-number of the count x in w is the same as the t-number of the count of x in v then either both w and v violate the constraint or neither does.*

Essentially, this theorem says constraints that are FO-definable with successor cannot distinguish among strings that are composed of the same number and type of substrings of some length k, where substrings can be counted only up to some threshold t.

We can use this theorem to show that *N..L is not FO definable with successor by presenting two strings which *N..L distinguishes but which are not distinguishable according to the criteria in Theorem 1. This would prove that *N..L is not LTT and thus not FO-definable with successor. Importantly, for any k and t we have to present two strings. These strings can depend on k and t.

We use notation a^k to mean the string consisting of k consecutive as. So $a^3 = aaa$. For any numbers k and t larger than 0, consider the words $w = a^k n a^k \ell a^k$ and $v = a^k \ell a^k n a^k$. Table 2.6 below shows the substrings

up to length k, and their number of occurrences. Each word has the same substrings and the same number of them. Note the left and right word boundaries ($\rtimes$ and $\ltimes$ respectively) are customarily included as part of the strings.

count	$w = \rtimes a^k n a^k \ell a^k \ltimes$	Notes
1	$\rtimes a^{k-1}$	
3	a^k	
1	$a^i n a^j$	(for each $0 \leq i, j \leq k-1$, $i + j = k - 1$)
1	$a^i \ell a^j$	(for each $0 \leq i, j \leq k-1$, $i + j = k - 1$)
1	$a^{k-1} \ltimes$	

count	$v = \rtimes a^k \ell a^k n a^k \ltimes$	Notes
1	$\rtimes a^{k-1}$	
3	a^k	
1	$a^i n a^j$	(for each $0 \leq i, j \leq k-1$, $i + j = k - 1$)
1	$a^i \ell a^j$	(for each $0 \leq i, j \leq k-1$, $i + j = k - 1$)
1	$a^{k-1} \ltimes$	

Table 2.5: The k-long substrings with their number of occurrences in the strings $w = a^k n a^k \ell a^k$ and $v = a^k \ell a^k n a^k$ with word boundaries.

As can be seen from the above table, the two strings have exactly the same number of occurrences of each k-long substring. Consequently, the t-numbers of each k-long substring is also the same. It follows, from Theorem 1 that these two strings cannot be distinguished by any constraint which is FO-definable with successor. More precisely, *any constraint which is FO-definable with successor is unable to distinguish in strings w and v whether n precedes ℓ or whether ℓ precedes n.* As such, no FO-definable constraint with successor can be violated by w but not by v and vice versa. It follows that *N..L is not FO definable with successor because for the reason that it this is precisely the distinction it makes.

Having established that linguistically motivated long-distance phonotactic constraints are not FO-definable with successor, we turn to the question of how such constraints can be defined from the logical perspective offered here. Essentially, there are two approaches. One is to increase the power of the logic. The other is to change the model—the representation—of strings. This section examines the first option and the next section examines the sec-

ond option. This interplay between logical power and representations and how it affects the expressivity of the linguistic system is a running theme of this book.

Monadic Second Order (MSO) logic is a logical language that is strictly more powerful than FO logic. Constraints that are MSO-definable with successor include every constraint which is FO-definable with successor because every sentence and formula in FO logic with successor is also a sentence and formula in MSO logic with successor and is interpreted in the same way. In addition to first order variables, MSO comes with **second order variables**. Generally, variables that are second order are allowed to vary over n-ary relations. The restriction to monadic second order variables means the variables in this logic can only vary over unary relations, which corresponds to *sets* of domain elements. This contrasts with first order variables, which recall vary only over elements of the domain.

MSO logic is defined formally in the appendix to Part I, so here we introduce it informally with examples. In MSO logic, the MSO variables are expressed with capital letters such as X, Y, and Z to distinguish them from first order variables which use lowercase letters like x, y, and z. Observe that $x \in X$ and $X(x)$ are synonyms. As with first order variables, second order variables are introduced into sentences and formula with quantifiers.

Additional Symbols in MSO logic	
X, Y, Z	variables which range over sets of elements of the domain
$x \in X$	checks whether an element x belongs to a set of elements X
$X(x)$	checks whether an element x belongs to a set of elements X

Table 2.6: Together with the symbols of FO logic shown in Table 2.2, these symbols make up MSO logic.

With MSO logic over successor, it is now possible to define the precedence relation as shown below.

$$\mathsf{closed}(X) \stackrel{\text{def}}{=} (\forall x, y)\big[(x \in X \wedge x \lhd y) \rightarrow y \in X\big] \tag{2.7}$$

$$x < y \stackrel{\text{def}}{=} (\forall X)\big[(x \in X \wedge \mathsf{closed}(X) \rightarrow y \in X\big] \tag{2.8}$$

Intuitively, a set of elements X in the domain of a model of some word w satisfies $\mathsf{closed}(X)$ only if every successor of every element in X is also in X. In short, $\mathsf{closed}(X)$ is true only for sets of elements X which are transitively

closed under successor. Then x precedes y only if for every closed set of elements X which x belongs to, y also belongs to X.

Figure 2.3 below illustrates these ideas. The successor model for the string $a\ell aana$ is shown. Six ellipses are shown, which represent the six nonempty sets of domain elements which are closed under successor and thus satisfy $\mathsf{closed}(X)$.

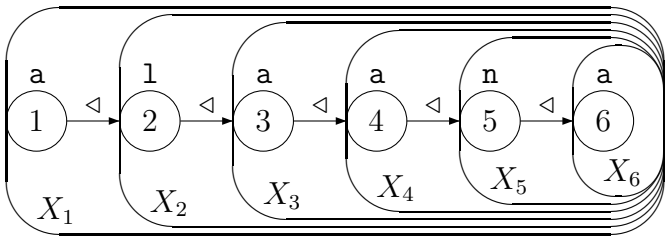

Figure 2.3: The successor model for the word $a\ell aana$. Rectangular regions indicate the sets of domain elements (X_i) which are closed under successor.

We can conclude that ℓ precedes n because every closed set which element 2 (which corresponds to ℓ) belongs to (X_1 and X_2) also includes the element 5 (which corresponds to n). Similarly, we can conclude that n does not precede ℓ because it is not the case that all closed sets which contain element 5 (which corresponds to n) also include element 2 (which corresponds to ℓ). Set X_4 for instance contains element 5 but not element 2.

Once the binary relation for precedence ($<$) has been defined, it is now straightforward to define the constraint *N..L with features.

$$\text{*N..L} \stackrel{\text{def}}{=} \neg(\exists x, y)[x < y \wedge \texttt{nasal}(x) \wedge \texttt{lateral}(y)] \tag{2.9}$$

The sentence above may look like a sentence of FO logic since no second order variables are present. However, it is important to remember that the precedence relation ($<$) is just an abbreviation for a longer formula, which is defined in MSO logic, and not within FO logic. Often whether a predicate is atomic or derived is not something that can be determined from inspecting a sentence or formula since the notation does not distinguish them. Usually one must be being acutely aware of the model signature to know whether a predicate is atomic or derived.

At this point, we have established that the linguistically motivated long-distance phonotactic constraint is not definable with FO logic with successor

but it is definable with MSO logic with successor. We thus ask: What other kinds of constraints are MSO-definable with successor?

Another constraint that is not FO-definable with successor but is MSO-definable constraint with successor is a constraint that requires words to have an even number of nasals. Words like *man* and *neonatology* obey this constraint since they have two nasals but words like *mannequin* and *nanotechnology* do not since they have three nasals.

To see that this constraint is not FO-definable with successor, we use Theorem 1 as before. For any nonzero numbers k and t, consider the words $w = a^k(na^k)^{2t}$ and $v = a^k(na^k)^{2t}na^k$. Observe that w obeys the constraint since it contains $2t$ nasals and $2t$ is an even number. On the other hand, v contains $2t + 1$ nasals and therefore violates the constraint. However, as Table 2.7 shows, these words have the same substrings of length k, and the same t-numbers of each substring.

$w = \rtimes a^k(na^k)^{2t}a^k \ltimes$			
count	t-number	k-long substrings	notes
1	1	$\rtimes a^{k-1}$	
$2t+2$	t	a^k	
$2t+2$	t	$a^i n a^j$	(for each $0 \leq i, j \leq k-1$, $i+j = k-1$)
1	1	$a^{k-1} \ltimes$	
$v = \rtimes a^k(na^k)^{2t}na^k \ltimes$			
count	t-number	k-long substrings	notes
1	1	$\rtimes a^{k-1}$	
$2t+2$	t	a^k	
$2t+2$	t	$a^i n a^j$	(for each $0 \leq i, j \leq k-1$, $i+j = k-1$)
1	1	$a^{k-1} \ltimes$	

Table 2.7: The k-long substrings and the t-numbers of their counts in $w = a^k(na^k)^{2t}$ and $v = a^k(na^k)^{2t}na^k$ with word boundaries.

However, this constraint is expressible with MSO logic with successor. We make use of some additional predicates, including general precedence ($<$) defined in Equation 2.8. The predicate firstN is true of x only if x is the first nasal occuring in the word (Equation 2.10). The predicate lastN is true of x only if x is the last nasal occuring in the word (Equation 2.11). Also,

two variables x and y stand in the $\lhd_{\mathsf{N}}$ only if y is the first nasal to occur after x (Equation 2.12). So $\lhd_{\mathsf{N}}$ is a a successor relation relativized to nasals.

$$\mathsf{firstN}(x) \stackrel{\text{def}}{=} \texttt{nasal}(x) \wedge \neg(\exists y)[\texttt{nasal}(y) \wedge y < x] \tag{2.10}$$

$$\mathsf{lastN}(x) \stackrel{\text{def}}{=} \texttt{nasal}(x) \wedge \neg(\exists y)[\texttt{nasal}(y) \wedge x < y] \tag{2.11}$$

$$x \lhd_{\mathsf{N}} y \stackrel{\text{def}}{=} \texttt{nasal}(x) \wedge \texttt{nasal}(y) \wedge x < y \wedge \neg(\exists z)[\texttt{nasal}(z) \wedge x < z < y] \tag{2.12}$$

Note we use the shorthand $x < y < z$ for $x < z \wedge z < y$.

With these predicates in place, we write Even-N as in Equation 2.13.

$$\textsc{Even-N} \stackrel{\text{def}}{=} (\exists X)\Big[\; (\forall x)[\mathsf{firstN}(x) \rightarrow X(x)] \wedge (\forall x)[\mathsf{lastN}(x) \rightarrow \neg X(x)] \; \Big] \wedge (\forall x, y)\big[x \lhd_{\mathsf{N}} y \wedge \big(X(x) \leftrightarrow \neg X(y)\big)\big] \tag{2.13}$$

In English, this says that a model of word w satisfies Even-N provided there is a set of domain elements X that includes the first nasal (if one occurs), does not include the last nasal (if one occurs) and for all pairs of successive nasals (if they occur), exactly one belongs to X. Consequently, words containing zero nasals satisfy the Even-N because the empty set of domain elements vacuously satisfies these three conditions. Words containing exactly one nasal do not satisfy Even-N because the first nasal and the last nasal are the same element x and it cannot both belong and not belong to X. However, words with exactly two nasals do satisfy Even-N because the first nasal belongs to X (satisfying the first condition), the last nasal does not (satisfying the second condition), and these two nasals are successive nasals and so are subject to the third condition, which they satisfy because exactly one of them (the first nasal) belongs to X. A little inductive reasoning along these lines lets one conclude that only words with an even number of nasals will satisfy Even-N as intended.

It is natural to wonder whether there is an abstract characterization of constraints that are MSO-definable with successor in the same way that Thomas (1982) provided an abstract characterization of constraints that are FO-definable with successor. In fact there is. Büchi (1960) showed that these constraints are exactly the ones describable with finite-state automata.

Theorem 2 (Characterization of MSO-definable constraints with successor)**.** *A constraint is MSO-definable with successor if and only if there is a finite-state automata which recognizes the words obeying the constraint.*

From the perspective of formal language theory, they are exactly the regular languages. Informally, these are formal languages for which the membership problem can be solved with a constant, finite amount of memory.

In this section we showed that FO-definable constraints with successor are not sufficiently powerful to express long-distance phonotactic constraints. One approach is to then increase the power of the logic. One logical system extends FO by adding quantification over monadic second order variables. This logic—MSO logic with successor—is able to express long-distance phonotactic constraints. However, MSO logic with successor also is also sufficiently expressive as a CDL to express constraints like Even-N.

Another way of putting it is like this. In the successor model, the information that in the word $a\ell aana$ the ℓ precedes the n is not immediately available from the representation. That information can be *deduced* but the deduction requires some computational effort. From the logical perspective taken here, this deduction requires MSO power and not FO power. Furthermore, once MSO power is admitted then it becomes possible to similarly deduce whether or not there are even numbers of elements with certain properties.

Another approach to developing a CDL which can express long-distance phonotactic constraints but not Even-N is to change the representation of strings; that is, to change the model signature. This is precisely the topic of the next section.

2.7 The Precedence Word Model

So far, the logics we have considered have been defined with respect to the successor model of words. However, as we have seen with phonological features vis a vis atomic letters, there are different models of strings. In this section, we consider the *precedence* model of strings. Simply, this model contains the precedence relation instead of the successor relation in its signature.

As with the successor model, there is a general construction for the determining the precedence model for any string. Given a string of w of length n the precedence model is constructed as follows. Since w is a sequence of n symbols, we let $w = a_1 a_2 \dots a_n$. Then set the domain $D = \{1, 2, \dots n\}$. For each symbol $a \in \Sigma$ and i between 1 and n inclusive, $i \in \mathsf{a}$ if and only if $a_i = a$.

And finally, for each i and j between 1 and n inclusive, the only elements of the precedence relation are (i, j) so long as $i < j$. This is summarized in Table 2.8. This construction guarantees the model's soundness: each string

D	$\stackrel{\text{def}}{=}$	$\{1, 2, \ldots n\}$
a	$\stackrel{\text{def}}{=}$	$\{i \in D \mid a_i = a\}$ for each unary relation a
$<$	$\stackrel{\text{def}}{=}$	$\{(i, j) \subseteq D \times D \mid i < j\}$

Table 2.8: Creating a successor model for any word $w = a_1 a_2 \ldots a_n$.

has a model and distinct strings will have distinct models.

Figure 2.4 shows the precedence model for the word *tent* in addition to a graphical diagram of it on its right.

$$
\begin{aligned}
&\mathcal{M}_{tent} \\
&= \langle D \mid \mathtt{t}, \mathtt{e}, \mathtt{n}, \mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}, < \rangle \\
&= \Big\langle \{1,2,3,4\} \mid \{1,4\}, \{2\}, \{3\}, \\
&\quad \varnothing, \varnothing, \ldots \varnothing, \\
&\quad \{(1,2),(1,3),(1,4), \\
&\quad\quad (2,3),(2,4),(3,4)\} \Big\rangle
\end{aligned}
$$

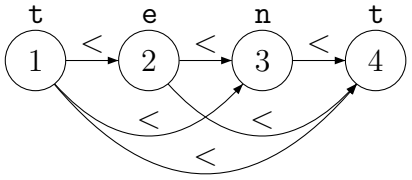

Figure 2.4: At left, the precedence model of the word *tent*. At right, a graphical diagram of this model.

The difference between the precedence model and the successor model is how the order of segments in the word are represented. In the precedence model, the fact that the n is preceded by t in the word *tent* is immediately available because the element corresponding to t is in the precedence relation with the element corresponding to the first t. Under the successor model, this information was not immediately available as it was not part of the representation. However, under the precedence model it is.

Take seriously from a psychological perspective, the precedence model can be taken to mean that as words are perceived, information about the precedence relations is being stored in memory as part of the lexical representation of the word.

Also, in the same way that we considered the successor model both with and without features, we can also consider a precedence model with and

without features. The precedence model introduced above was without features, but it is a simple matter to replace the unary relations in that model with the ones in Table 2.4.

It is straightforward to now write the constraint *N..L in the CDL which we call "FO with precedence with features."

$$\texttt{*N..L} \stackrel{\text{def}}{=} \neg\exists x, y(x < y \wedge \texttt{nasal}(x) \wedge \texttt{lateral}(y)) \tag{2.14}$$

Equation 2.14 looks identical to Equation 2.9. However, there is critical difference. In Equation 2.14, the precedence relation is an atomic formula but in Equation 2.9 it is a user-defined predicate in MSO logic.

It is natural to ask of course whether a constraint like *NT is expressible in this CDL. The answer is Yes because successor is FO-definable from precedence. Equation 2.15 shows how. Essentially, x is succeeded by y only if x precedes y and there is no element z such that $z < y$ and $x < z$.

$$x \lhd y \stackrel{\text{def}}{=} x < y \wedge \neg(\exists z)[x < z < y] \tag{2.15}$$

It is a striking fact that successor is FO-definable from precedence but precedence is MSO-definable from successor. This is a considerable asymmetry between the successor and precedence models of strings.

One important consequence is that the CDL "MSO with precedence" we is equivalent in expressive power to the CDL "MSO with successor" discussed in the previous section. This is because with MSO logic, precedence can be defined from successor as shown previously. Likewise because MSO logic properly extends FO logic, successor can also be defined from precedence. So at the level of MSO, these two models make no distinctions among the kinds of constraints that can be expressed. Constraints in each CDL correspond to exactly the regular stringsets.

This asymmetry also implies by Theorem 1 that there are constraints which are FO-definable with precedence but not FO-definable with successor. Heinz *et al.* (2011) provide a linguistically motivated example based on long-distance dissimilation.

Long-distance dissimilation in phonology is typically when one speech sound x changes because there is another similar speech sound y in the word, though y may be non-adjacent to x (Bennett, 2013). To illustrate we present an example from Latin. The adjectival suffix /-alis/ is realized as [aris] if the stem includes an [l]. For example, [episcop-alis] 'episcopal' but [lun-aris] 'lunar'. If these kinds of examples are viewed as constraints on surface forms,

then this generalization could be expressed in English as follows. Two [l]s are permitted within a word only if [r] occurs between them. We acknowledge there is debate regarding the robustness of the generalization we present and readers are referred to Bennett (2013, Chapter 8) for a clear survey of the issues. Nonetheless, because this constraint "Lateral Dissimilation" is a linguistically-motivated constraint, it suffices for our purposes. It can be expressed precisely in FO logic with precedence as follows.

$$\mathsf{LD} \stackrel{\text{def}}{=} (\forall x, y)\Big[\big(x \neq y \wedge \texttt{lateral}(x) \wedge \texttt{lateral}(y)\big) \rightarrow (\exists z)[\texttt{rhotic}(z) \wedge x < z < y]\Big] \tag{2.16}$$

It is easy to show that this constraint is not FO definable with successor by Theorem 1. Intuitively, this is because we can construct two words within which the order of [l] speech sounds and [r] speech sounds cannot be determined. More specifically, for any numbers k and t, we can find two words w and v which are indistinguishable in terms of the t-numbers of the counts of the k-long substrings. These two words are $w = a^k\ell a^k r a^k \ell a^k r a^k$ and $v = a^k\ell a^k \ell a^k r a^k r a^k$. Clearly, w obeys the constraint LD but v does not. In model-theoretic terms, the precedence model of w satisfies LD, but the precedence model of v does not. Table 2.7 show the counts of the k-long substrings of these two words with word boundaries. Because every k-long substring in w occurs exactly the same number of times in v and vice versa, the t-numbers of these counts will be the same as well. Consequently, by Theorem 1, the set of words satisfying LD is not FO-definable with successor.

There is also an abstract characterization of the FO-definable constraints with precedence due to McNaughton and Papert (1971).

Theorem 3 (Characterization of FO-definable constraints with precedence)**.** *A constraint is FO-definable with precedence if and only if there is a number n such that for all strings x, y, z if xy^nz obeys the constraint then for all $k > n$, xy^kz obeys the constraint too.*

This characterization says that FO-definable constraints with precedence can only distinguish iterations within strings up to some finite n. Two strings xy^iz and xy^jz with both $i, j > n$ but $i \neq j$ cannot be distinguished by any FO-definable constraint with precedence. As McNaughton and Papert (1971) amply document, there are other independently-motivated characterizations of this class as well.

count	$w = a^k\ell a^k r a^k \ell a^k r a^k$	Notes
1	$\rtimes a^{k-1}$	
5	a^k	
2	$a^i r a^j$	(for each $0 \leq i, j \leq k-1$, $i + j = k - 1$)
2	$a^i \ell a^j$	(for each $0 \leq i, j \leq k-1$, $i + j = k - 1$)
1	$a^{k-1} \ltimes$	

count	$v = a^k\ell a^k \ell a^k r a^k r a^k$	Notes
1	$\rtimes a^{k-1}$	
5	a^k	
2	$a^i r a^j$	(for each $0 \leq i, j \leq k-1$, $i + j = k - 1$)
2	$a^i \ell a^j$	(for each $0 \leq i, j \leq k-1$, $i + j = k - 1$)
1	$a^{k-1} \ltimes$	

Table 2.9: The k-long substrings with their number of occurrences of the strings $w = a^k\ell a^k r a^k \ell a^k r a^k$ and $v = a^k\ell a^k \ell a^k r a^k r a^k$ with word boundaries.

The above characterization can be used to show that EVEN-N is not FO-definable with precedence. Again, the strategy is to consider any n and then to find strings w, v, x, y, z and numbers $i, j > n$ such that $w = xy^i z$ and $v = xy^j z$ where EVEN-N distinguishes w and v in the sense that one violates EVEN-N and the other does not. If the constraint were FO-definable with precedence such strings could not exist by Theorem 3. In this case, one solution is to set $x = z = \lambda$ (the empty string), $y = ma$, $i = 2n$ and $j = 2n + 1$. Then $w = (ma)^{2n}$ and $v = (ma)^{2n+1}$. Clearly, w has an even number of nasals since it has $2n$ [m]s but v has an odd number since it has $2n+1$ [m]s. Thus EVEN-N distinguishes these strings and thus by Theorem 3 it cannot be FO-definable with precedence.

In this section, we considered a model of words where order is represented with the precedence relation instead of the successor relation. It was shown that long-distance constraints can readily be expressed in the CDL "FO with precedence." Furthermore, local phonotactic constraints like *NT can also be expressed because successor is FO-definable from precedence. However, the converse is not true. This asymmetry means that FO with precedence is strictly more expressive than "FO with successor." It was also shown that EVEN-N is not expressive in this system. Finally, it was noted that "MSO with precedence" is equally expressive as "MSO with successor". Once there

is MSO power, successor and precedence are each definable from the other.

More generally, this section established the following. Although one way to increase the expressivity of a CDL is to increase the power of the logic, another way is to change the representations underlying the models. This speaks directly to the interplay between representations and computational power, one of the themes of this chapter.

We conclude that the only CDL discussed so far that can express both local and long-distance phonotactic constraints (like *NT and *N..L) and fails to express constraints like Even-N is the CDL "FO with precedence."

2.8 Propositional-style Logic

While the CDL "FO with precedence" appears sufficient to describe both local and long-distance phonotactic constraints, it is natural to wonder whether weaker logical systems suffice as well. One clue that FO is more expressive than necessary, is that it is straightforward to define constraints that are sensitive to the number of occurences of a structure in a word. This counting is in fact part of the abstract characterization of "FO with successor" in Theorem 1.

For example, Equation 2.5 gave a definition for the constraint *NT. It is very easy to write a similar constraint that only penalizes words three NT sequences but not two as shown below.

$$\begin{aligned} *\mathsf{3NT} \stackrel{\text{def}}{=} \neg(\exists x_1, x_2, x_3 x_4, x_5, x_6)\Big[& \\ & \big(x_1 \lhd x_2 \wedge \texttt{nasal}(x_1) \wedge \texttt{voiceless}(x_2)\big) \\ \wedge & \big(x_3 \lhd x_4 \wedge \texttt{nasal}(x_3) \wedge \texttt{voiceless}(x_4)\big) \\ \wedge & \big(x_5 \lhd x_6 \wedge \texttt{nasal}(x_5) \wedge \texttt{voiceless}(x_6)\big) \\ \wedge & \big(x_1 \neq x_3 \wedge x_1 \neq x_5 \wedge x_3 \neq x_5\big)\Big] \end{aligned} \tag{2.17}$$

Note we use $x \neq y$ as shorthand for $\neg(x = y)$. According to this constraint hypothetical words like *kampantasank* are ill-formed, but words like *kampantasak* are well-formed. Is there a principled way to eliminate this kind of counting from the CDLs?

There is, and this is precisely what this section accomplishes. Propositional logic is a logical system that is weaker than FO. In this section we motivate and define a propositional-style logic by restricting FO logic to only

