MSO	*N..L, Even-N	Even-N
FO	*NT	*NT, *N..L
	$\triangleleft$	$<$

Figure 2.5: Classifying the constraints *NT, *N..L, and Even-N.

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.

$$\begin{aligned}\mathsf{*3NT} \overset{\text{def}}{=} \neg(\exists x_1, x_2, x_3 x_4, x_5, x_6)\Big[& \\ & \big(x_1 \triangleleft x_2 \wedge \mathtt{nasal}(x_1) \wedge \mathtt{voiceless}(x_2)\big) \\ & \wedge \big(x_3 \triangleleft x_4 \wedge \mathtt{nasal}(x_3) \wedge \mathtt{voiceless}(x_4)\big) \\ & \wedge \big(x_5 \triangleleft x_6 \wedge \mathtt{nasal}(x_5) \wedge \mathtt{voiceless}(x_6)\big) \\ & \wedge \big(x_1 \neq x_3 \wedge x_1 \neq x_5 \wedge x_3 \neq x_5\big)\Big]\end{aligned} \tag{2.16}$$

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. We do this for both the successor and the precedence models of strings.

The resulting CDLs do not have the ability to count in the manner above. More generally, the abstract characterizations of the resulting CDLs corresponds to a particular type of memory model with clear cognitive implications. We return to these broader issues after introducing this weaker propositional logic.

Sentences of propositional logic are Boolean combinations of **atomic propositions**. The Boolean connectives, presented in Table 2.2, are the symbols: $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), $\rightarrow$ (implication), and $\leftrightarrow$ (biconditional).[1] Classically, the atomic propositions can be anything from sentences like "All men are mortal" to "The sample contained chlorine." The truth of any sentence in propositional logic can be computed from the truth values of the atomic propositions and the standard ways the Boolean connectives are interpreted. Furthermore, one way to interpret the meaning of a sentence ϕ in propositional logic is as the set of worlds in the universe for which ϕ would evaluate to `true`.

Here, the universe is the set of possible strings Σ^* and each string in Σ^* is a "world" in this universe. Thus, each propositional sentence ϕ will pick out some set of strings in Σ^*, which are those for which it can be said to be true of.

What are the atomic propositions in this universe of strings? They are going to be model-theoretic *connected* structures. The idea is that each such structure S is true of any string w whose model M_w contains it. In order to be precise, we must introduce the meanings of *connected*, *restriction*, and *contains*.

We begin with what is meant by *connected*. For each structure $\mathcal{S}$ with relational model signature $\mathcal{M} = \langle D \mid R_1, \ldots R_n \rangle$ let the binary relation C be defined as follows.

$$C \stackrel{\text{def}}{=} \big\{(x, y) \in D \times D \mid \exists i \in \{1 \ldots n\}, \exists (x_1 \ldots x_m) \in R_i, \\ \exists s, t \in \{1 \ldots m\}, x = x_s, y = x_t \big\}$$

Further, let C^* denote the transitive closure of C (so C^* is the least set which contains C and for which it is the case that whenever $(x, y) \in C^*$ and

[1] In technical presentations of propositional logic, only $\wedge$ (conjunction) and $\neg$ (negation) are included as primitives since the others can be derived either rom $\wedge$ (conjunction) and $\neg$ (negation) or from $\vee$ (disjunction) and $\neg$ (negation).

$(y, z) \in C^*$ then $(x, z) \in C^*$ too). Then a structure A is *connected* if for all x, y in the domain of A, $(x, y) \in C^*$.

As an example, consider the structure M_{abbcc} in the standard successor model. This is a connected structure because every pair of elements is related by the successor relation. In particular, that domain elements 1 and 4 are connected is witnessed by these elements of the successor relation $(1, 2), (2, 3), (3, 4)$. In fact, the structure of every string under every model discussed is connected under this definition.

What is an example of an unconnected structure? Under the signature $\langle D \mid \triangleleft, a, b, c\rangle$, consider $A = \{\{1, 2\} \mid \emptyset, \{1\}, \{2\}, \emptyset\}$. This structure contains two elements (one is a and one is b) but they are not connected by any series of relations.

Next we discuss what it means for one structure to be a *restriction* of another. Structure $A = \langle D^A \mid R_1^A, \ldots R_n^A\rangle$ *is a restriction of* structure $B = \langle D^B \mid R_1^B, \ldots R_n^B\rangle$ (denoted $A \sqsubseteq B$) if $D^A \subseteq D^B$ and for each m-ary relation R_i, we have $R_i^A = \{(x_1 \ldots x_m) \in R_i^B \mid x_1, \ldots, x_m \in D^A\}$. So A is essentially what is left of B after it is stripped of elements and relations which are not wholly within the domain of A.

Finally, we say structure $A = \langle D^A \mid R_1^A, \ldots R_n^A\rangle$ *is contained by* $B = \langle D^B \mid R_1^B, \ldots R_n^B\rangle$ structure (denoted $A \sqsubseteq B$) if A is connected and A isomorphic to a restriction of B. Whenever $A \sqsubseteq B$, we also say A is a *factor* of B.

As an example consider the successor model with features and the structure with domain $D = \{1, 2\}$, the successor relation given by $\{(1, 2)\}$, with `nasal`$=\{1\}$, `voiceless`$=\{2\}$, and with all other unary relations denoting phonological features equal to the empty set. This connected structure, which we denote `NT`, represents a nasal immediately succeeded by a voiceless segment. Compare this structure with $\mathcal{M}_{tent}$ in the successor model with features presented in Figure 2.2. Structure `NT` is contained by $\mathcal{M}_{tent}$; equivalently, `NC` is a factor of $\mathcal{M}_{tent}$.

For each string $w \in \Sigma^*$, let $F(\mathcal{M}_w)$ denote the set of factors of the structure $\mathcal{M}_w$ and let $F_k(\mathcal{M}_w)$ be set of factors whose size is less than or equal to k (recall that the size of a structure is equal to the cardinality of its domain). Formally, $F(w) = \{\mathcal{S} \sqsubseteq M_w\}$ and $F_k(w) = \{\mathcal{S} \sqsubseteq M_w, |\mathcal{S}| \leq k\}$.

Finally, we "lift" the definition of F and F_k to sets of strings as follows.

$$F(S) = \bigcup_{w \in S} F(\mathcal{M}_w) \tag{2.17}$$

$$F_k(S) = \bigcup_{w \in S} F_k(\mathcal{M}_w) \tag{2.18}$$

In the propositional-style logic we present here, the atomic propositions are elements of $F(\Sigma^*)$. The semantics of this logic are straightforward. A model $\mathcal{M}$ satisfies an atomic proposition $\mathcal{S}$ provided $\mathcal{S}$ is a factor of $\mathcal{M}$.

$$\mathcal{M} \models \mathcal{S} \text{ iff } \mathcal{S} \sqsubseteq \mathcal{M} \tag{2.19}$$

Finally we note that every propositional sentence ϕ can be associated with the size of its largest atomic proposition. In other words, the

The remainder of the logic is defined like every other propositional logic. Sentences of propositional logic combine atomic propositions with the Boolean connectives ($\wedge$ conjunction, $\vee$ disjunction, $\neg$ negation, $\rightarrow$ implication, and $\leftrightarrow$ biconditional), and these combinations have their usual meanings. The language associated with a propositional sentence ϕ is also defined in the usual manner.

$$L(\phi) = \{w \in \Sigma^* \mid \mathcal{M}_w \models \phi\} \tag{2.20}$$

As was the case with FO and MSO logics, this propositional-style logic we have introduced depends on a model signature. This is because what the the atomic propositions — the connected structures — are depend on the model signture.

In the remainder of this section we discuss four CDLs: PROP($\triangleleft$), PROP($<$), PROP($\triangleleft, \rtimes, \ltimes$), and PROP($\triangleleft, \rtimes, \ltimes$, features). These refer to this propositional-style logic the model signatures with successor, precedence, successor with word boundaries and successor with word boundaries and phonological features, respectively.

Each of these four CDLs has a very similar characterization as expressed in the following theorem.

Theorem 4 (Characterization of PROP-definable constraints)**.** *A constraint is PROP-definable with if and only if there is a number k such that for any two strings w and v, whenever the model structures of these two strings have the same factors up to size k (so $F_k(w) = F_k(v)$ under the given model), then either both w and v violate the constraint or neither does.*

That the theorem is true is not hard to see. If two strings w, v have exactly the same k-factors then models satisfy exactly the same set of atomic propositions (which are the k-factors they contain). Since the truth of any propositional formula ϕ depends only on the truth or falsity of its atomic propositions, it must be the case that either both $\mathcal{M}_w \models \phi$ and $\mathcal{M}_v \models \phi$ or neither $\mathcal{M}_w$ nor $\mathcal{M}_v$ satisfy ϕ.

Significant literature exists on the classes of formal languages definable with some of these CDLs. In particular the class of formal languages definable with PROP($\triangleleft$) are Strongly Locally Testable (**?**). The class of formal languages definable with PROP($<$) are Piecewise Testable (**?**).

Much literature has also been devoted to the class of constraints definable with PROP($\triangleleft, \rtimes, \ltimes$). This model extends the successor model left and right word boundaries. The signature of this model is $\langle D \mid \triangleleft, \rtimes, \ltimes, (\mathtt{a})_{a\in\Sigma}\rangle$ where symbols $\rtimes$ and $\ltimes$ denote unary relations which are interepreted as the left and right word boundaries, respectively. The model-theoretic representation of a string $w = a_1 a_2 \dots a_n$ is presented in Table 2.9.

D	$\overset{\text{def}}{=}$	$\{0, 1, 2, \dots n+1\}$
$\mathtt{a}$	$\overset{\text{def}}{=}$	$\{i \in D \mid a_i = a\}$ for each unary relation $\mathtt{a}$
$\triangleleft$	$\overset{\text{def}}{=}$	$\{(i, i+1) \subseteq D \times D\}$
$\rtimes$	$\overset{\text{def}}{=}$	$\{0\}$
$\ltimes$	$\overset{\text{def}}{=}$	$\{n+1\}$

Table 2.9: Creating a successor model with word boundaries for any word $w = a_1 a_2 \dots a_n$.

For example, Figure 2.6 shows the structure the successor model with word boundaries assigns to the string *tent*.

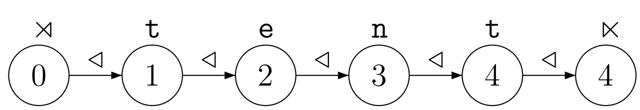

Figure 2.6: A graphical depiction of the successor model with word boundaries of the word *tent*.