Acquiring Digital Phonology

Erik F. Tjong Kim Sang[1]
Alfa-informatica
University of Groningen
erikt@let.rug.nl

Abstract

In the phonological feature theory presented in "Phonological Networks" by D.G. Gilbers language specific digital phonology (DIGPHON) models are constructed by carefully modifying a universal model. I present a learning strategy that can be used for deriving language specific DIGPHON models from a universal model. The strategy amounts to recognizing obligatory constraints on the combinations of articulatory features present in a language. The constraints can be used to convert a universal DIGPHON model into a language specific model mechanically.

1 Introduction

In his thesis "Phonological Networks" D.G. Gilbers presents a digital feature theory of phonology. The human articulatory system is represented by a model containing two components: the F-component and the C-component. The F-component is a system of hierarchically organized articulatory features. It models the human articulatory capabilities, disabling non-sensical combinations of articulatory features. The C-component consists of a system of switches and connections. The position of the switches (the state of the C-component) determines the speech sound the model will produce at a moment. The articulatory system of a six-month old child is considered to be universal and capable of producing every speech sound present in human languages. When child learns to speak the language of its parents, his articulatory system restricts itself to the speech sounds present in his language. In the digital feature theory of Gilbers this restriction is modeled by an evolution of the C-component.

Gilbers presented a universal model for vowel production with a few language specific models. The language specific models are constructed by hand, making sure all phonological principles defined by Gilbers are satisfied by the model. This construction process is non-trivial. Yet the human speech production system is capable of deriving a language specific C-component from the universal one. I am interested in finding out what learning functions can be used to model this derivational process.

In this paper I will describe a learning strategy that can be used to derive language specific models of digital phonology (DIGPHON) from a universal DIGPHON model. In the examples used here I will restrict myself to models of vowel production. The techniques described here can also be used for a future DIGPHON model for consonant

[1]This paper presents the stage of research in acquisition in the Digital Phonology framework in October 1992. I should mention that since then Gilbers renamed the framework "Phonological Networks". I am grateful to Dicky Gilbers for introducing me to DIGPHON and to him and Gosse Bouma for critical comments on earlier versions of this paper. Of course, remaining errors and peculiarities remain my responsibility. This research was granted by the Dutch Foundation for Language, Logic and Information.

production. I start with a short description of the universal vowel model described in [Gilbers 92]. After this, I present a learning strategy for deriving language specific constraints. The constraints are based on the *optional implications* defined in [Gilbers 92]. The outline of the learning strategy will be followed by a discussion of the phonological principles DIGPHON models have to satisfy. After that, I present results which were achieved by using the strategy.

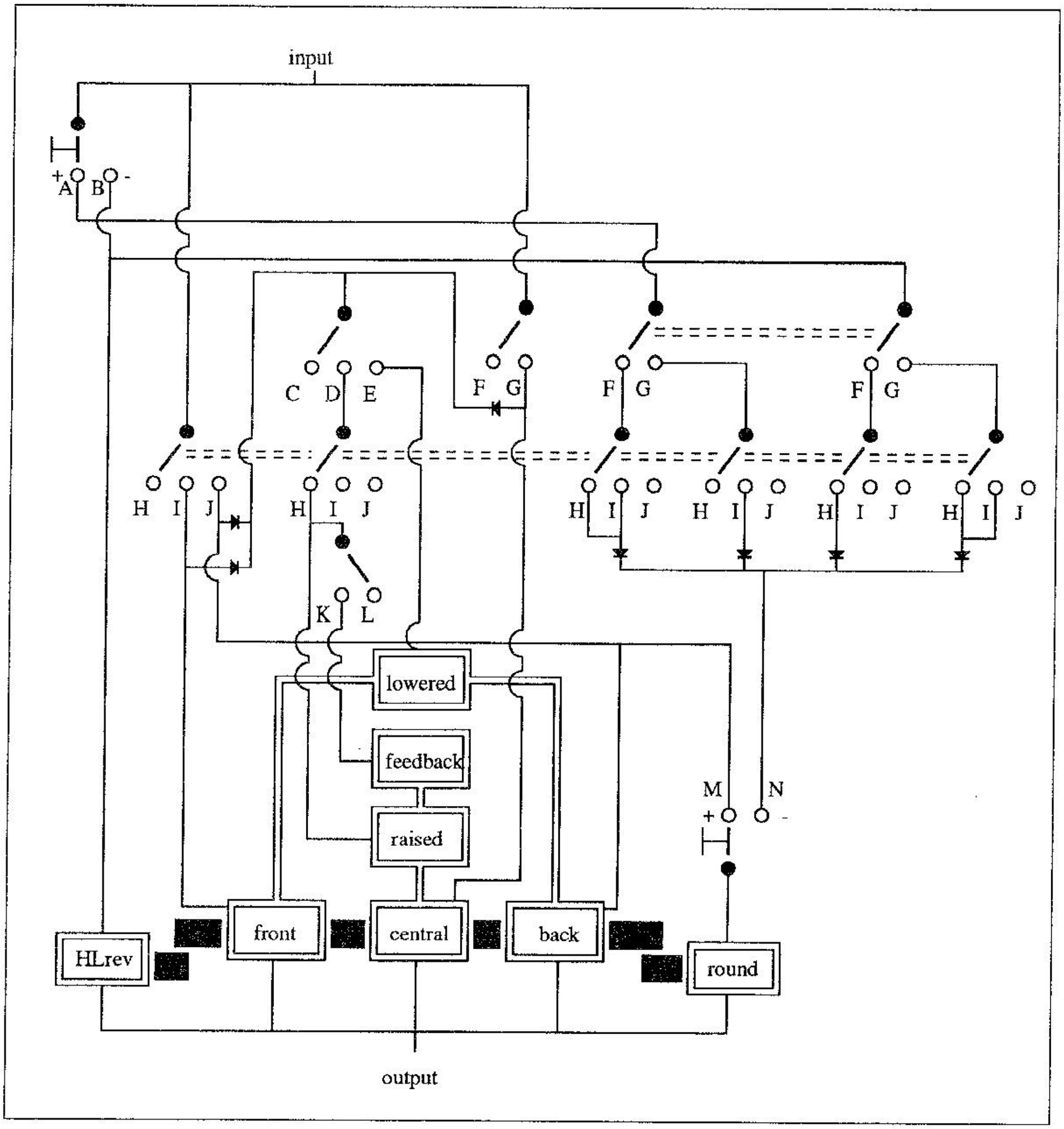

Figure 1: The universal vowel model presented in [Gilbers 92].

2 The universal vowel model

The universal vowel model developed in [Gilbers 92] is drawn in figure 1. It consists of connections, diodes, feature boxes and different types of switches: binary switches ($switch_{FG}$ and $switch_{KL}$), ternary switches ($switch_{CDE}$ and $switch_{HIJ}$) and polarity switches ($switch_{AB}$ and $switch_{MN}$). Switch dependency is denoted by dotted lines and similar switch position names. The model contains eight articulatory features: the three carriers: *front*, *central* and *back*, and the five modulators: *lowered*, *raised*, *feedback*, *round* and *HLrev*. A feature is considered on if it is connected with the input. For example, if $switch_{AB}$ is in position B the feature *HLrev* is connected with the input and is considered on. If $switch_{AB}$ is in position A, the feature *HLrev* cannot be connected to the input so *HLrev* will be off. Modulators that are connected to more basic features (placed at a lower position in figure 1) with double lines are exceptions to this procedure. These modulators can only be on, if at least one of the features they are connected to is on too. For example, *lowered* will only be on if it is connected with the input and if *front* or *back* is on. I will use the notation [$switch_{XY}$:Y] for describing that $switch_{XY}$ is in position Y and the notation [$feature_i$:*1*] ([$feature_i$:*0*]) for describing that $feature_i$ is on (off).

I will use the term *state* of the model (conform [Bouma 91]) to refer to the positions of the switches of the model at a particular moment. For instance, if the state of the model is such that [$switch_{FG}$:G] and all other switches are in the position shown in figure 1, feature *central* will be on and all other features will be off. The feature list {*central*} will be produced. I will represent feature combinations as ordered lists of binary values using the order *front*, *central*, *back*, *lowered*, *raised*, *feedback*, *round*, *HLrev*. For example, the representation for the feature combination consisting of *central* alone is $[0, 1, 0, 0, 0, 0, 0, 0]$ /a/.

Each DIGPHON model imposes an order on the sounds it produces. In the DIGPHON vowel models all vowels receive a value, the markedness value. This value indicates the effort that is needed to produce the vowel. The more effort is needed to produce a vowel, the higher its markedness value will be. In DIGPHON models the markedness value is the sum of some penalty values. The penalties of a vowel can be computed by examining the resources that are needed to produce it. The resources receive a penalty according to the table in figure 2. For example, producing $[0, 1, 0, 0, 0, 0, 0, 0]$ /a/ in the universal vowel model requires the usage of one binary switch ($switch_{FG}$) for a passive vowel containing one carrier (*central*), resulting in a markedness value of 1+1=2.

Complexity	Contrast		Penalty
	passive	active	
uses 1 carrier	1	4	1 or 4
uses 2 carriers	4	1	1 or 4
uses binary switches			1 per switch
uses ternary switches			2 per switch
automatical modulation	1	0	1 or 0
dependent place modulation			4 per switch
uses the *round* reverse switch ($switch_{MN}$)			8
uses the *HLrev* switch ($switch_{AB}$)			10

Figure 2: Markedness-value assignment table for vowels.

I will now explain the penalty assignment procedure in short. This explanation is not necessary for understanding the remainder of this paper. The larger part of figure 2 should explain itself. Vowel contrast is considered to be passive if none of the features *lowered*, *raised* and *feedback* is on. Otherwise vowel contrast is considered to be active. Automatical modulation amounts to a extra penalty if *back* is activated. Dependant place modulation concerns the three switches $switch_{CDE}$, $switch_{HIJ}$ and $switch_{KL}$ (in that order) in the paths to *lowered*, *raised* and *feedback*. When a current flows through these switches the vowel produced receives an extra penalty for every used switch. The reader may wish to verify that producing $[0,0,1,1,0,0,0,1]$ /ɥ/ requires two ternary switches and both the *round* reverse switch ($switch_{MN}$) and the *HLrev* switch ($switch_{AB}$), combined with one dependent place modulation switch for a active vowel containing one carrier, which results in a markedness value of 2*2+8+10+4+4=30.

This possibility of computing markedness values from effort involved in producing sounds, is one of the most interesting aspects of the DIGPHON models.

3 Constraints for language-specific models

The easiest way to derive a language specific DIGPHON model from a universal DIGPHON model is by limiting the positions switches can take. By doing this, connections and switches can be removed from the model. This approach is successful for deriving models for two of the four vowel patterns Gilbers discussed in his chapter 4: the Abkhaz vowel pattern and the most common vowel pattern. Not every vowel pattern can be derived this way. Most language specific models use switches and connections which are not present in the universal model drawn in figure 1. Gilbers states that this universal model is incomplete. It also contains hidden paths, called *optional implications*, that are necessary for deriving language specific models (see [Gilbers 92], section 4.2.1). The optional implications do not change the behavior of the universal model. Figure 3 contains an example of a hidden connection becoming visible in a language specific model. A hidden binary switch is present between the two inputs of $switch_{MN}$. The default position of this switch is open but in some language specific models the switch is closed, thus creating a connection.

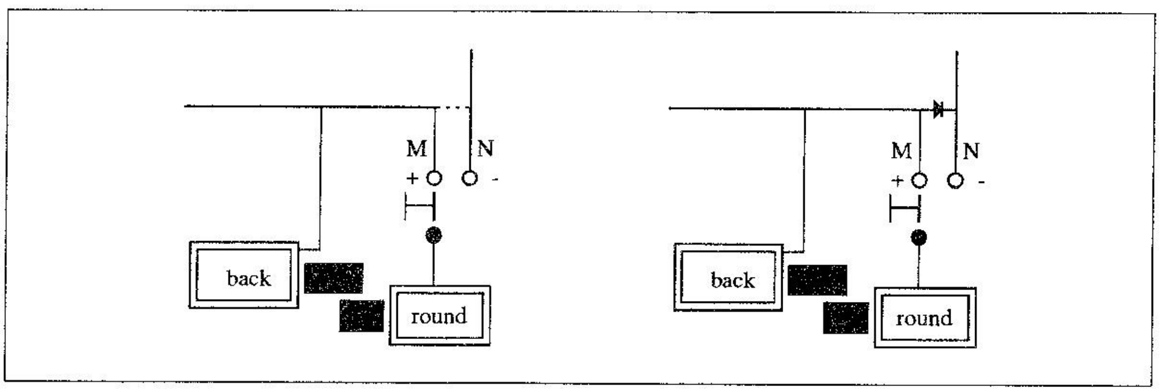

Figure 3: An example of an hidden connection becoming visible in a language specific model. To the left a part of the universal vowel model. To the right the corresponding part of the Dutch vowel model.

I will use the optional implications of Gilbers for deriving language specific models. My goal is to define a language specific model as a combination of a universal model and a list of optional implications. The optional implications will work like constraints on the universal model.

3.1 Optional implications

Gilbers assumed that three optional implication types are necessary in a universal DIGPHON model. I will use a logical representation for the implications:

A1. $[carrier : value_1] \rightarrow [modulator : value_2]$
for instance $[central : 0] \rightarrow [raised : 0]$.

A2. $[carrier_1 : value_1] \rightarrow [carrier_2 : value_2]$
for instance $[back : 1] \rightarrow [front : 0]$.

A3. $[modulator_1 : value_1] \rightarrow [modulator_2 : value_2]$
for instance $[feedback : 1] \rightarrow [raised : 1]$.

Not every implication listed above is possible in the universal vowel model. The implications should not make the universal model produce feature combinations that are unpronounceable. By using the following definitions we can formalize this idea:

definition 1. An implication $\mathcal{I}$ ($X \rightarrow [feature_i : value_j]$) is ALLOWED for a set of patterns $\mathcal{C}$ when every pattern $\mathcal{P}_k \epsilon \mathcal{C}$ either satisfies $\mathcal{I}$ or can be converted into a pattern $\mathcal{P}_l \epsilon \mathcal{C}$ by changing the value of $feature_i$ into $value_j$.

definition 2. An implication $\mathcal{I}$ is EMBEDDED in a set of patterns $\mathcal{C}$ in case every pattern $\mathcal{P} \epsilon \mathcal{C}$ satisfies $\mathcal{I}$.

The optional implications will change the behavior of the model. Only optional implications that are allowed for the set of sensical feature combinations can be part of a universal DIGPHON model. Other implications will make the universal model produce non-sensical feature combinations. We can use the implications that are embedded in a set of language specific feature combinations as a description of the language.

3.2 Necessity of other optional implications types

The three implication types Gilbers mentioned in his thesis, are insufficient to account for every language specific model. For instance, the Japanese vowel pattern consists of:

1. $[0, 0, 0, 0, 0, 0, 0, 0]$ /ə/
2. $[1, 0, 0, 0, 0, 0, 0, 0]$ /i/
3. $[1, 1, 0, 1, 0, 0, 0, 0]$ /ɛ/
4. $[0, 1, 0, 0, 0, 0, 0, 0]$ /a/
5. $[0, 1, 1, 1, 0, 0, 1, 0]$ /ɔ/
6. $[0, 0, 1, 0, 0, 0, 0, 0]$ /ɯ/

A language specific model for Japanese should be able to produce $[0, 1, 1, 1, 0, 0, 1, 0]$ /ɔ/ but not $[0, 1, 1, 1, 0, 0, 0, 0]$ /ʌ/. From the implications mentioned in A1, A2 and A3 eleven implications can be used for making the model behave like that. The first seven deny $[0,1,1,1,0,0,0,0]$ by requiring *round* should be on if one of the other features has the same value as in the feature combination. The implications 8 to 11 deny the feature combination by requiring that one of the other modulators should have a different value when *round* is off.

1. $[front : 0] \rightarrow [round : 1]$ also rejects $[0,0,0,0,0,0,0,0]$ /ə/
2. $[central : 1] \rightarrow [round : 1]$ also rejects $[1,1,0,1,0,0,0,0]$ /ε/
3. $[back : 1] \rightarrow [round : 1]$ also rejects $[0,0,1,0,0,0,0,0]$ /ɯ/
4. $[lowered : 1] \rightarrow [round : 1]$ also rejects $[1,1,0,1,0,0,0,0]$ /ε/
5. $[raised : 0] \rightarrow [round : 1]$ also rejects $[0,0,0,0,0,0,0,0]$ /ə/
6. $[feedback : 0] \rightarrow [round : 1]$ also rejects $[0,0,0,0,0,0,0,0]$ /ə/
7. $[HLrev : 0] \rightarrow [round : 1]$ also rejects $[0,0,0,0,0,0,0,0]$ /ə/
8. $[round : 0] \rightarrow [lowered : 0]$ also rejects $[1,1,0,1,0,0,0,0]$ /ε/
9. $[round : 0] \rightarrow [raised : 1]$ also rejects $[0,0,0,0,0,0,0,0]$ /ə/
10. $[round : 0] \rightarrow [feedback : 1]$ also rejects $[0,0,0,0,0,0,0,0]$ /ə/
11. $[round : 0] \rightarrow [HLrev : 1]$ also rejects $[0,0,0,0,0,0,0,0]$ /ə/

Unfortunately every implication rejects at least one of the feature combinations present in the Japanese vowel pattern. We can see that no optional implication mentioned in A1, A2 and A3 will able a language specific vowel model to accept all Japanese vowels and reject $[0,1,1,1,0,0,0,0]$ /ʌ/. A remedy for this is changing the set of possible optional implications. I propose to use the next three implications instead of the implications A1, A2 and A3:

B1. $true \rightarrow [feature : value]$ in which *feature* can be either *carrier* or *modulator*. This implication results in restricting the value of *feature* to one value.

B2. $[carrier : value_1] \rightarrow [feature : value_2]$ (a combination of A1 and A2)

B3. $[carrier : value_1]\&[feature : value_2] \rightarrow [modulator : value_3]$

Implication B3 creates the possibility of using the implication *[central:1]*&*[back:1]* → *[round:1]*. This implication will reject $[0,1,1,1,0,0,0,0]$ (ʌ) without rejecting any feature combination present in the Japanese vowel pattern. By using this implication we are able to create a language specific vowel model that produces all Japanese vowels and no more. I assume that the universal C-component for vowel production consists of the model Gilbers put forward (see figure 1) combined with optional implications that are of the formats mentioned in B1, B2 and B3 and that are allowed for the set of pronounceable combinations of articulatory features. The only reason I can give for choosing exactly these implications, is that they have proved to be necessary and sufficient to describe language models of vowel production I have examined so far. I believe that they will be sufficient for describing any other language specific DIGPHON model and I hope that many switch models for non-sensical sets of combinations of articulatory features need implications that are not of the formats I chose.

3.3 Deriving language specific obligatory implications

We now have optional implications with sufficient power for deriving language specific DIGPHON models from a universal model. Deriving these language specific models amounts to recognizing what optional implications in the universal vowel model are obligatory in the language specific model[2]:

definition 3. All implications that are allowed for some universal set of feature combinations and are embedded in the set of feature combinations present in the language $\mathcal{L}$ are OBLIGATORY in the language specific DIGPHON model for $\mathcal{L}$.

hypothesis 1. If we have a DIGPHON model $\mathcal{M}$ and:

1. $\mathcal{M}$ produces all and only feature combinations present in the universal vowel set.
2. $\mathcal{M}$ only produces feature combinations that satisfy the implications that are of one of the formats B1, B2 and B3 and that are obligatory for a sensible language $\mathcal{L}$.
3. Feature combinations that are produced by $\mathcal{M}$ are not restricted by implications other than implications that are derivable from the ones mentioned in 2.

then $\mathcal{M}$ will produce all and only feature combinations of language $\mathcal{L}$.

I believe that humans, after a development stage during the first six months of their life, have a vowel production system that resembles the universal vowel model of [Gilbers 92] combined with optional implications of the formats described in B1, B2 and B3. Acquiring a language specific vowel model amounts to *recognizing the implications that are obligatory in the language specific model.* By putting constraints on the format of the implications, I put constraints on the language specific models that can be derived from the universal vowel model. Hence the phrase 'sensible language' in hypothesis 1. I cannot guarantee that a model can be derived from the universal vowel model for any arbitrary set of sensible feature combinations. I hope the constraints I put on the format of the implications will prevent us to derive sets of feature combinations that do not occur in any human language.

4 Defining DIGPHON vowel models

In the previous section I presented a strategy that can be used for deriving obligatory implications for language specific vowel models. It is interesting to try to use these implications for deriving language specific DIGPHON models. In most cases it is not very difficult to come up with a model consisting of switches and connections that is capable of producing all and only the feature combinations that occur in a specific language. However, this model will not always be a DIGPHON model. Gilbers has

[2]The DIGPHON feature representation of the vowels contains six vowels with two possible representations (first mentioned in [Kas 89]). Deciding which one of these representations to use in a language depends on characteristics of other vowels in the language. This decision will influence the optional implications needed for modeling a language. I will avoid the problem of deciding which of two equivalent representations is most suitable in a language and simply choose the least marked one, the one with *[HLrev:0]*.

devised a number of limitation principles DIGPHON models should satisfy. In this section I will discuss these principles.

The first two principles are:

principle 1. The Natural Limitation Principle. Feature combinations which are logically possible but – for physiological reasons – unpronounceable, ought to be ruled out by the model ([Gilbers 92], page 19).

principle 2. The Basic Modulation Principle. In case a modulator depends on another feature (carrier or modulator, the dependency is represented by a double line in figure 1) the modulator cannot become active without the feature being active. ([Gilbers 92], page 52).

These two principles limit the number of vowels a DIGPHON model is able to produce. The universal vowel model satisfies these two principles. A language specific vowel model will violate at least one of these principles if it is able to produce a feature combination the universal vowel model cannot produce. We will have to force language specific vowel models to produce only feature combinations that can be produced by the universal vowel model. We can do this by forcing the language specific vowel models only to produce feature combinations that satisfy the implications that are embedded in the universal vowel model.

There is one more principle a DIGPHON model should satisfy. The vowel derivation of the model should confirm with the markedness assignment theory Gilbers described in his section 5.2.3. This markedness assignment theory consists of a procedure that assigns a markedness value to the feature combinations. This produces a hierarchy of feature combinations that should confirm with the general theory of markedness of vowels (see section 2).

The way Gilbers formulated this extra principle made it hard to verify. The only way to check that the vowel derivation in a language specific vowel model confirms with the markedness theory, is computing the markedness values of all vowels that can be produced by the model. If this results in rejection of the model, there is no obvious way of repairing the model. I attempt to formalize the markedness constraint with four principles. It can easily be checked that a model satisfies the extra principles. If one of them rejects a model, the cause of rejection is obvious from the definition of the principle and the model can be repaired.

principle 3a. In case a language $\mathcal{L}$ contains feature combinations containing *[HLrev:1]*[3]:

1. The DIGPHON model for $\mathcal{L}$ should contain the *HLrev* switch ($switch_{AB}$).
2. All states of the DIGPHON model for $\mathcal{L}$ that produce a feature combination containing *[HLrev:1]* should contain the *HLrev* switch in negative polarity position.
3. For every feature combination containing *[HLrev:0]* there should exist a state in the DIGPHON model for $\mathcal{L}$ that produces the combination and that contains the *HLrev* switch in positive polarity position.

[3]Every language specific model is capable of producing at least one feature combination that contains *[HLrev:0]* and satisfies *[back:1]* $\Leftrightarrow$ *[round:1]*: [0, 0, 0, 0, 0, 0, 0, 0] /ə/. [Gilbers 92], page 130 predicts that this vowel is universal.

principle 3b. In case a language $\mathcal{L}$ contains feature combinations that do not satisfy the equivalence relation [*back:1*] $\Leftrightarrow$ [*round:1*][3]:

1. The DIGPHON model for $\mathcal{L}$ should contain the *round* reverse switch ($switch_{MN}$)
2. All states of the DIGPHON model for $\mathcal{L}$ that produce a feature combination that does not satisfy this relation, should contain the *round* reverse switch in negative polarity position.
3. For every feature combination in $\mathcal{L}$ satisfying the relation, the DIGPHON model for $\mathcal{L}$ should contain a state that produces the feature combination and that contains the *round* reverse switch in positive polarity position.

principle 3c. In case both 3a and 3b are applicable to a language $\mathcal{L}$, for every feature combination in $\mathcal{L}$ containing $[HLrev : 0]$ and satisfying equivalence relation $[back : 1] \Leftrightarrow [round : 1]$, a DIGPHON model for $\mathcal{L}$ should contain a state that produces the feature combination and that contains both the *HLrev* switch and the *round* reverse switch in positive polarity position.

principle 4. In case a language $\mathcal{L}$ contains categories containing *raised:1*, all paths to *raised* in a model for $\mathcal{L}$ should contain at least two switches.

The principles 3a, 3b, 3c and 4 are necessary to make sure the models will confirm with the markedness assignment theory. They will make sure that the feature combinations will receive the correct penalties for using reverse switches (see section 2). A model that has been derived from the universal vowel model and that confirms with the principles 3a, 3b, and 3c, is a DIGPHON vowel model. The four principles 3a, 3b, 3c and 4 I mentioned here are the only markedness related restrictions I will impose on language specific DIGPHON vowel models[4]. I will make sure that the language specific DIGPHON models satisfy the Natural Limitation Principle and the Basic Modulation Principle by deriving the models from a correct universal DIGPHON model in such a way that the principles are still satisfied in the derived model.

5 Experimental results

I looked at the possibility of generating language specific vowel models mechanically. First I wrote a program which generated language specific obligatory implications of the formats B1, B2 and B3. The strategy the program uses consists of three steps. First it generates all possible optional implications of the formats B1, B2 and B3. Then it extracts from these the set of implications which are embedded in the target language and finally it removes all redundant implications from this set. I applied this program to a corpus of 32 vowel patterns. The corpus contains the four vowel patterns discussed in the fourth chapter of [Gilbers 92] and an additional 28 vowel patterns randomly chosen from [Maddieson 84]. For every one of these vowel patterns the program was able to generate a set of implications which were sufficient to describe the vowel pattern (see

[4]One of the interesting facts of DIGPHON models is that markedness values of sounds are a result of the effort that has to be taken to produce the sounds. The extra principles I presented in this section restrict the way a DIGPHON model produces specific sounds. In a way this will make the markedness values of sounds model-independent which will reduce the value of the models.

Language	Implication		
	B1	B2	B3
Abkhaz	$front = 0$ $central = 1$ $back = 0$ $lowered = 0$ $raised = 0$ $round = 0$		
Dutch	$feedback = 0$ $HLrev = 0$	$[back : 1] \rightarrow [round : 1]$	$[front : 0]\&[back : 0] \rightarrow [round : 0]$ $[front : 0]\&[central : 0] \rightarrow [lowered : 0]$ $[central : 0]\&[lowered : 1] \rightarrow [round : 0]$
most common	$lowered = 0$ $raised = 0$ $HLrev = 0$	$[back : 1] \rightarrow [round : 1]$ $[back : 0] \rightarrow [round : 0]$	
Japanese	$raised = 0$ $HLrev = 0$	$[central : 0] \rightarrow [lowered : 0]$ $[central : 1] \rightarrow [lowered : 1]$ $[central : 0] \rightarrow [round : 0]$ $[back : 0] \rightarrow [round : 0]$	$[central : 1]\&[back : 1] \rightarrow [round : 1]$
Russian	$raised = 0$	$[front : 0] \rightarrow [lowered : 0]$ $[front : 0] \rightarrow [HLrev : 0]$ $[central : 0] \rightarrow [lowered : 0]$ $[central : 1] \rightarrow [HLrev : 0]$ $[back : 0] \rightarrow [round : 0]$ $[back : 1] \rightarrow [round : 1]$	$[front : 1]\&[central : 1] \rightarrow [lowered : 1]$ $[central : 1]\&[lowered : 0] \rightarrow [round : 1]$
Zulu	$raised = 0$ $HLrev = 0$	$[central : 0] \rightarrow [lowered : 0]$ $[central : 1] \rightarrow [lowered : 1]$ $[back : 0] \rightarrow [round : 0]$ $[back : 1] \rightarrow [round : 1]$	

Figure 4: Non-redundant obligatory implications for six language specific vowel models.

figure 4). This result supports the claim of hypothesis 1; the implication formats B1, B2 and B3 are sufficient to describe language specific vowel patterns.

After this I made a program which uses the language specific obligatory implications for generating language specific DIGPHON models. The program transforms the universal vowel model to a model represented in logic, adds the implications defining the target language to that model and transforms the logical model back to a language specific switch model. I again applied this program to the corpus of 32 vowel patterns. It generated valid DIGPHON models for all 32 languages. For 24 languages a straightforward application of logical rewrite rules was sufficient for generating a valid DIGPHON model. The models the program generated for the remaining eight languages (Ao, Carib, Dutch, Hebrew, Japanese, Maori, Nunggubuyu and Thai) contained markedness errors, related to violations of principle 3a, 3b and 4. These errors can be fixed by adding and removing switches from the language specific models. The error-fixing process proved to be non-trivial and I cannot guarantee that the routine I use at this moment will be successful for other languages.

Apart from generating 28 new language specific vowel models, the model generation program actually managed to improve one of the vowel models presented in [Gilbers 92]. From the four vowel models mentioned in the fourth chapter of [Gilbers 92] three are identical to the ones generated by the program. One model differed: the model of the Dutch vowel pattern. An analysis of the Dutch model of Gilbers has shown that it does

Abkhaz	2	most common	6	Lappish	7	Silha	4
Alabama	4	Dutch	13	Maasai	10	Sudanese	7
Albanian	7	Greek	6	Maori	6	Swahili	6
Ao	7	Haida	4	Mura	4	Tagalog	3
Aranda	4	Hebrew	6	Nunggubuyu	4	Thai	8
Basque	6	Japanese	6	Quileute	5	Totonac	4
Carib	7	Katcha	8	Romanian	6	Vietnamese	12
Cayapa	5	Klamath	5	Russian	6	Zulu	6

Figure 5: The languages in the corpus with the number of vowels in each language.

not satisfy the markedness principles (in fact, it contains some redundant paths). The model presented in figure 6 does satisfy the markedness principles. Therefore it is the correct DIGPHON model for the Dutch vowel pattern.

6 Concluding remarks

In this paper I have outlined a learning strategy that can be used to derive language specific DIGPHON models from a universal model. The strategy amounts to recognizing obligatory language specific constraints on a universal model. These constraints have to satisfy two rules: they should be of the format specified in section 3.2 and they should be embedded in the set of language specific feature combinations (see section 3.1). I used this strategy together with a reformulation of the markedness constraint for generating 32 language specific DIGPHON models.

The learning task modeled here is in principle simple: based on the knowledge which combinations of articulatory features are and which are not in a language, a model needed to be derived that produces all and only the feature combinations present in the language. This is a language learning task that makes use of both positive and (implicit) negative information about elements present in the language. According to elementary learning theory developing a model that decides if a feature combination is in the language, is a trivial task ([Pinker 79], [Gold 67]). However, the markedness constraint on the DIGPHON models makes the task of deriving language specific DIGPHON models non-trivial.

It is difficult to imagine that the derivational plan used in the second program described in section 5 is cognitively valid. People probably do not use logical rewrite rules and peculiar switch additions when they acquire a vowel pattern. In that sense the plan seems of few value for modeling the acquisition of the human speech production system. However I believe that the knowledge that the acquisition process can be performed mechanically, combined with the constraints that need to be imposed on this process, is of relevance to future models of phonological acquisition. The strategy developed here can be used to generate a large collection of language specific DIGPHON models. The optional implications used in these models can be used for the development of complete universal DIGPHON models that allow a cognitively valid acquisition process for the human phonological system.

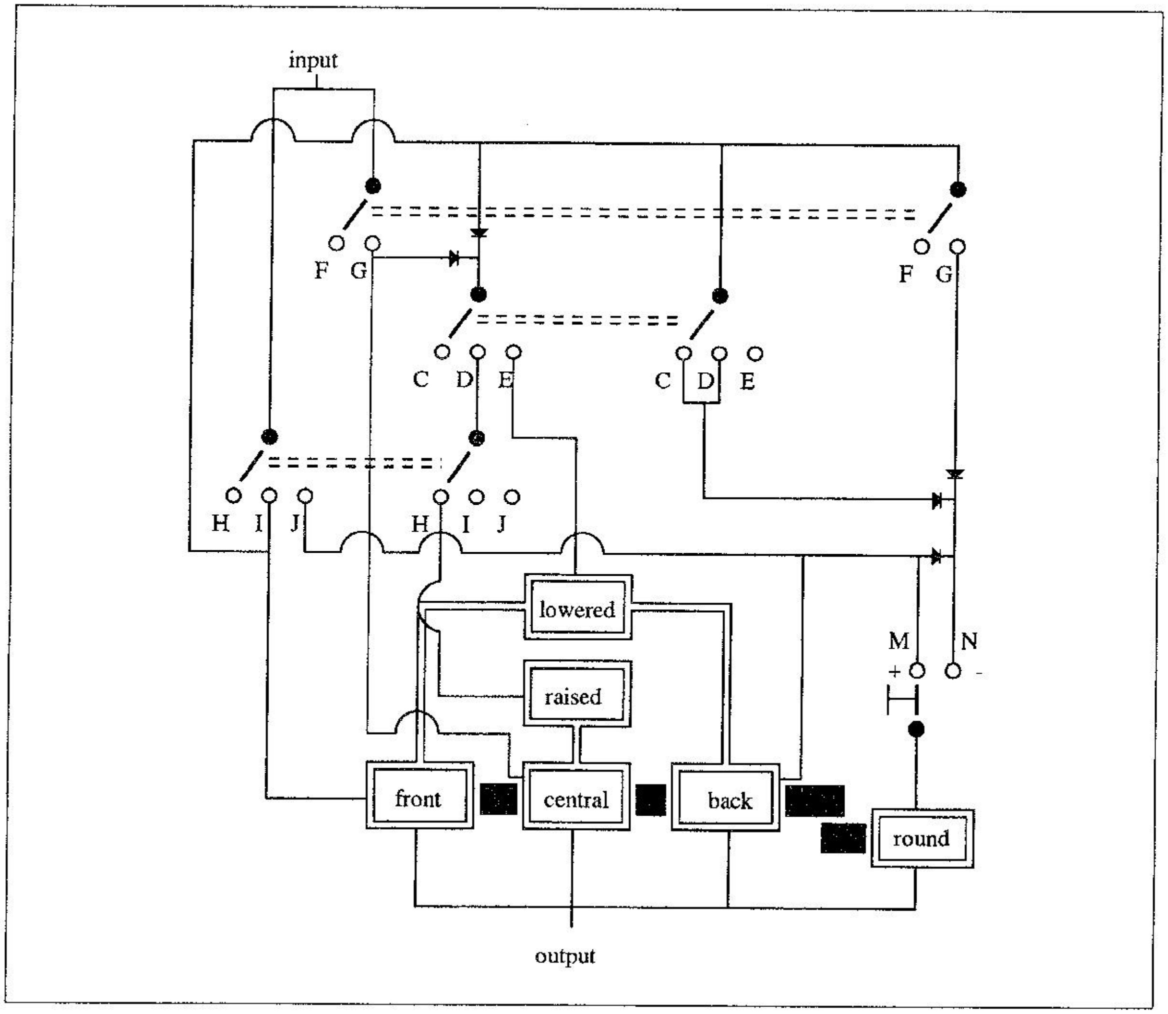

Figure 6: The model for Dutch generated by the strategy described in this section.

References

[Bouma 91] Gosse Bouma. A logical reconstruction of digital phonology. In Steven Bird, editor, *Declarative Perspectives on Phonology*, Edinburgh Working Papers in Cognitive Science. University of Edinburgh, 1991.

[Gilbers 92] D.G. Gilbers. *Phonological Networks.* PhD thesis, University of Groningen, 1992. ISSN 0928-0030.

[Gold 67] E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[Kas 89] Mark Kas. *De fonologie de computer in: een testprogramma.* (Dutch). unpublished manuscript, University of Groningen, 1989.

[Maddieson 84] Ian Maddieson. *Patterns of sounds.* Cambridge University Press, 1984.

[Pinker 79] Steven Pinker. Formal models of language learning. *Cognition*, 7:217–283, 1979.