

# An augmented chart data structure with efficient word lattice parsing scheme in speech recognition applications

Lee-Feng Chien, Lin-Shan Lee

*Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, ROC*

Keh-Jiann Chen

*The Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC*

Received 19 September 1989

Revised 11 January 1991

**Abstract.** This paper proposes an augmented chart data structure with an efficient word lattice parsing scheme in speech recognition. The augmented chart and the associated parsing algorithm can represent and very efficiently parse, without changing the fundamental principles of chart parsing, a lattice of lexically highly ambiguous word hypotheses in speech recognition. Every word lattice can be mapped to the augmented chart, with the ordering and link between the word hypotheses being well preserved in the augmented chart. A jump edge is defined in order to link edges representing word hypotheses physically separate, but connectable from a practical point of view. Preliminary experimental results show that with augmented chart parsing, all the possible constituents of the input word lattice can be constructed and no constituent needs to be built more than once. This significantly reduces computational complexity, especially when serious lexical ambiguity exists in the input word lattice as in the case of many speech recognition problems. This augmented chart parsing is thus a very useful and efficient approach to language processing problems in speech recognition.

**Zusammenfassung.** In diesem Artikel wird ein verbesserter Datentabellenaufbau mit besonderer Wortvernetzung für die Anwendung in der automatischen Spracherkennung vorgeschlagen. Die verbesserte Strukturierung und der dazugehörige Suchalgorithmus gliedern ein Gitter von Worthypothesen mit einem hohen Grad lexikalischer Ambiguität sehr effizient, ohne jedoch die grundsätzlichen Prinzipien des Tabellen-Parsing zu ändern. Jedes Wortgitter kann auf die um Ordnungs- und Beziehungsverwandtschaften ergänzten Tabellen abgebildet werden, wobei die Reihenfolgen und Beziehungen der Worthypothesen in den erweiterten Tabellen enthalten sind. Die Außenzone ist so definiert, daß Ränder verbunden werden können, die physikalisch getrennt sind, aber vom praktischen Gesichtspunkt aus verbunden werden dürfen. Erste experimentelle Ergebnisse haben gezeigt, daß mit der neuen Tabellenstruktur alle möglichen Bestandteile der eingegebenen Wortgitter konstruiert werden können und kein Bestandteil mehr als einmal gebildet werden muß. Dies reduziert den Rechenaufwand erheblich, besonders wenn starke lexikalische Ambiguitäten in den eingegebenen Wortgittern vorkommen. Daher ist die verbesserte Tabellengliederung ein sehr hilfreicher und effizienter Schritt zur Lösung von Problemen, wie sie bei der automatischen Sprecherkennung auftreten.

**Résumé.** Ce document propose une structure des données sous forme de tableau, enrichie d'une grille de mots et d'un système d'analyse grammaticale exploitables dans le domaine de la reconnaissance de la parole. Ce tableau enrichi et l'algorithme d'analyse grammaticale qui y est associé peuvent représenter et analyser grammaticalement de façon très efficace une grille d'hypothèses de mots émis dans le domaine de la reconnaissance de la parole avec un niveau d'ambiguïté lexicale élevé, sans pour autant changer les principes fondamentaux de l'analyse grammaticale sous forme étendue. Chaque grille de mots peut être entrée avec ordre et cohérence dans le tableau enrichi parmi des hypothèses de mots bien mémorisées dans ledit tableau. Un point de rupture est déterminé par rapport à des points de liaison que représentent des hypothèses de mots séparés dans le texte mais qu'il est en réalité possible de relier. Les résultats expérimentaux préliminaires nous montrent que l'enrichissement de l'analyse grammaticale sous forme de diagramme étendu permet l'interprétation de tous les composants possibles à l'entrée du réseau lexical et élimine la nécessité de créer chaque composant plus d'une fois. Cela réduit la complexité des calculs de manière significative surtout en cas d'ambiguïté lexicale importante dans l'entrée de la grille lexicale comme cela arrive dans de nombreux problèmes de reconnaissance de la parole. Cet enrichissement de l'analyse grammaticale sous forme de tableau représente une approche très utile et très efficace en ce qui concerne les problèmes de traitement de l'information dans le domaine de la reconnaissance de la parole.

**Keywords.** Word lattice parsing, chart parsing, speech recognition.

## 1. Introduction

In this paper, conventional chart data structure has been augmented for efficient word lattice parsing so as to handle the high degree of ambiguities encountered in speech recognition applications. A word lattice is a set of word hypotheses produced by an acoustic signal processor in continuous speech recognition applications and may include problems such as word boundary overlapping, lexical ambiguities, missing or extra phones, recognition uncertainty, errors, etc. The purpose of parsing such a word lattice is to obtain, efficiently and accurately, the most promising candidate sentence at an acceptable computational complexity by means of grammatical constraints and appropriate data structure design. For example, in the process of continuous speech recognition, very often it happens that not only more than one word may be produced for a given segment of speech (such as homonyms, especially for languages with large number of homonyms such as Chinese (Lee and Tseng, 1987)), but that many competing word hypotheses can be produced at overlapping, adjoining or separate segments of the acoustic signal without there being any set of aligned word boundaries. This will result in a huge number of sentence hypotheses, each formed by a combination of a sequence of word hypotheses, so that the exhaustive parsing of all these sentence hypotheses with a conventional text parser is computationally inefficient or even prohibitively difficult. Examples of possible ap-

proaches to solving this problem include developing ways of first finding the most promising sentence hypotheses for parsing, with the intermediate parsing results shared by different sentence hypotheses with some common parts, etc. In any case, a really efficient approach is desired. The abstract diagram in Figure 1 demonstrates such a problem. The concept of word lattice parsing seems to be a very attractive strategy for such a problem, as is shown in Figure 2.

Several algorithms for parsing such word lattices have been proposed (Tomita, 1986; Chow and Ronkos, 1989). These algorithms have been shown to be very efficient in parsing less ambiguous natural languages such as English obtained in speech recognition. However, they are all strictly left-to-right parsing algorithms, thus with relatively limited applications for cases in which other strategies such as island-driven or even right-to-left are more useful (Hayes et al., 1986; Huang and Shiu, 1988), e.g. for corrupted word lattices with extra, missing or erroneous phones in speech recognition (Ward et al., 1988). For example, it has been found that in speech recognition, the replacement of left-to-right parsing techniques by center-out techniques working from word hypotheses with a high certainty value is very helpful (Hayes et al., 1986), and some people even claim that Chinese sentence parsing is best performed from right to left due to the head-final structures of Chinese (Huang and Shiu, 1988). It has also been found that island-driven parsing algorithms are specially valuable in speech recognition appli-

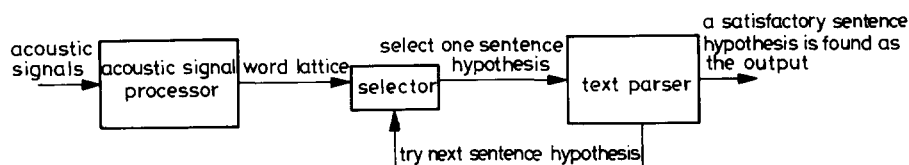


Fig. 1. A text parser used in a speech recognition system.

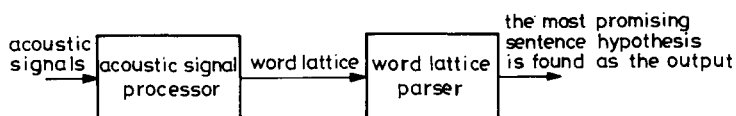


Fig. 2. A word lattice parsing scheme in speech recognition application.

cations because they can function more gracefully when not all the correct words of an utterance are obtained from the acoustic signal processor (Ward et al., 1988). On the other hand, the chart is an efficient working structure widely used in many natural language processing systems and has been shown to be a very effective approach (Kay, 1980; Thompson, 1981), it is basically designed to parse a sequence of fixed and known words instead of an ambiguous word lattice. In this paper, the conventional chart is successfully extended or augmented in such a way that it is able to represent a word lattice. Firstly, an input word lattice for parsing can be represented by the augmented chart through a mapping procedure. In this chart, a minimum number of vertices is used to indicate the end points of all the word hypotheses in the lattice, and an inactive edge is used to represent each word hypothesis. Each of these edges starts at the vertex indicating the beginning of the word hypothesis and terminates at the vertex indicating the end. Also, specially designed jump edges are constructed to link some edges whose corresponding word hypotheses can possibly be connected, but themselves are physically separate in the chart. In this way the basic operation of a chart parser can be properly performed for a word lattice. Note that in such a scheme, every constituent (edge) is constructed only once, regardless of the fact that it may be shared by many different sentence hypotheses. On the other hand, the conventional functions, operations and properties of a chart parser as well as some useful extensions such as the use of lexicalized grammars and island driven parsing, are not affected by the augmentation at all. Therefore the augmented chart parsing proposed in this paper is a very efficient and attractive parsing scheme for many language processing problems in speech recognition applications. A word lattice parser based on the augmented chart data structure proposed here has been implemented and tested for Chinese as part of a long term project to develop an efficient Mandarin speech recognition system (Lee et al., 1987; Chien et al., 1990; Chien, 1991), and the preliminary results are very encouraging.

In the following, Section 2 formalizes the word lattice parsing problem and Section 3 introduces

the concept of the augmented chart. The procedure for mapping an input word lattice to the augmented chart is described in Section 4, while some properties of the mapping procedure are discussed in Section 5. The parsing scheme and the parsing structures are then presented in Sections 6 and 7, and some preliminary experimental results, further extensions and a discussion of a different application example feature in Sections 8, 9 and 10 respectively. Some final comments are given in Section 11.

## 2. The problem definition

*The problem:* Given a word lattice  $W = \{w_1, \dots, w_n\}$  and a grammar  $G$ , the problem is to efficiently and accurately find out all grammatically valid sentence hypotheses and their sentence structures based on  $W$  and  $G$ <sup>1</sup>.

Some related definitions are given below.

*Word lattice.* A word lattice  $W$  is a partially ordered set of word hypotheses,  $W = \{w_1, \dots, w_m\}$ , where each hypothesis  $w_i$ ,  $i = 1, \dots, m$ , is characterized by *begin*, the beginning point, *end*, the end point, *cat*, the category, *phone*, the associated phonemes, and *name*, the name of the word hypothesis. These word hypotheses are sorted in the order of their end-points; that is, for every pair of word hypotheses  $w_i$  and  $w_j$ ,  $i < j$  implies  $\text{end}(w_i) \leq \text{end}(w_j)$ . As an example, a sample word lattice constructed for demonstration purposes is shown in Figure 3; we will use this word lattice as an example throughout this paper.

*Starting word.* A word hypothesis  $w_i$  in  $W$  is a starting word if there is no other word  $w_j$  in  $W$  such that  $\text{end}(w_j) \leq \text{begin}(w_i)$ . For example, the starting words in the sample word lattice in Figure 3 are  $w_1$  and  $w_2$ .

*Ending word.* A word hypothesis  $w_i$  in  $W$  is an ending word if there is no other word  $w_j$  in  $W$  such that  $\text{end}(w_i) \leq \text{begin}(w_j)$ . For example, the ending words in the sample word lattice in Figure 3 are  $w_5$  and  $w_6$ .

*Connected words.* Let  $\leq$  be a defined partial ordering on  $W$ , that is,  $w_i \leq w_j$  iff  $\text{end}(w_i) \leq$

<sup>1</sup> Here we assume that, in practical applications, every word hypothesis in the word lattice is reliable enough.

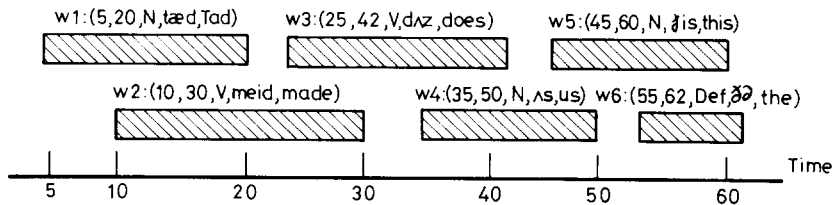


Fig. 3. A sample word lattice showing a set of overlapped word hypotheses which are assumed to be produced by an acoustic signal processor in speech recognition. Each rectangle denotes the time segment of the acoustic signal for the word hypothesis and above it is the 5-tuple information, from left to right, i.e., *begin, end, cut, phone* and *name*, respectively.

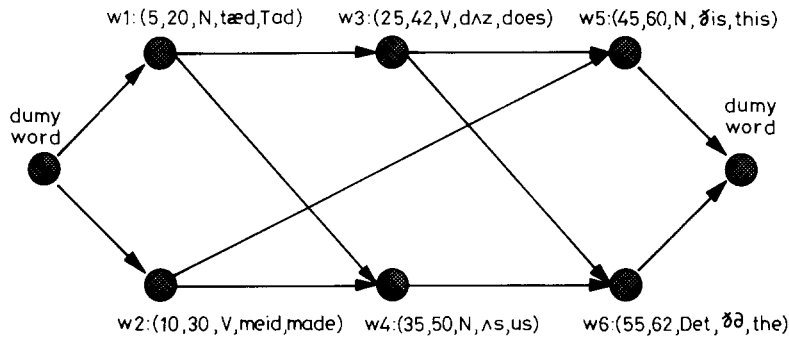


Fig. 4. The link between the word hypotheses in the sample word lattice.

$\text{begin}(w_j)$ ; otherwise  $w_i \leq w_j$ . The link between any pair of word hypotheses  $w_i$  and  $w_j$  is then defined as the following:  $\text{WConn}(w_i, w_j) = \text{true}$ , or  $w_i$  and  $w_j$  are said to be connected iff  $i < j$ ,  $w_i \leq w_j$ , and there is no other word hypothesis  $w_k$ , so that  $w_i \leq w_k \leq w_j$ ; otherwise  $\text{WConn}(w_i, w_j) = \text{false}$ . In other words,  $w_i$  and  $w_j$  are connected if there is no other word hypothesis located exactly between the boundaries of the two words. In Figure 4 network is used to depict this link for the above example, where every word hypothesis is represented by a circle and every pair of circles representing word hypothesis  $w_i, w_j$  with relation  $\text{WConn}(w_i, w_j) = \text{true}$  is connected by an arrow pointing from circle  $w_i$  to circle  $w_j$ . For example, in the above word lattice,  $\text{WConn}(w_3, w_6) = \text{true}$  but  $\text{WConn}(w_1, w_6) = \text{false}$ .

**Sentence hypothesis.** A sequence of word hypotheses  $S_i, S_i = w_{i1}, w_{i2}, \dots, w_{ir}$ , is a sentence hypothesis if it satisfies the following conditions:

- (i)  $w_{i1}$  is a starting word,
- (ii)  $w_{ir}$  is an ending word,

- (iii) for every pair of  $w_{ik}$  and  $w_{i(k+1)}$ , where  $1 \leq k < r - 1$ ,  $w_{ik} \leq w_{i(k+1)}$  and  $\text{WConn}(w_{ik}, w_{i(k+1)}) = \text{true}$ .

Moreover, a sentence hypothesis  $S_i$  is grammatically valid only if it can be generated by a grammar. For instance, a sample of simplified rule-based grammar<sup>2</sup> is given in Figure 5, and all the possible sentence hypotheses of the sample word lattice are listed in Figure 6. However, according to the grammar in Figure 5, only the first sentence in Figure 6 denoted by “\*” is grammatically valid.

### 3. The augmented chart

The conventional chart parsing algorithm is designed to parse a sequence of words. In this section the chart is augmented for parsing word lattices. The augmented chart is a directed unicyclic

<sup>2</sup> In this paper the grammar discussed is in fact not limited to a rule-based grammar. In Section 9 the use of the lexicalized grammar will also be explained.

r1. S->NP VP  
 r2. NP-> Det N  
 r3. NP-> N  
 r4. VP-> V  
 r5. VP-> V N

Fig. 5. The sample grammar.

1. Tad does this.
2. Tad does the.
3. Tad us the.
4. made us the.
5. made this

Fig. 6. All the possible sentence hypotheses obtained from the sample word lattice.

graph specified by a two-tuple  $\langle V, E \rangle$ , where  $V$  is a sequence of vertices and  $E$  is a set of edges. Each vertex in  $V$  represents the end-point of some of the word hypotheses in the input word lattice, while the edge set is divided into three disjoint groups: inactive, active and jump edges.

An inactive edge is a data structure to represent a complete constituent. It is characterized by the following information: *from*, the vertex where the edge starts (the begin vertex), *to*, the vertex where the edge ends (the end vertex), *cat*, the associated category, *sub-inactive*, the list of the immediately spanned inactive edges included and *name*, the word name (for lexical edges only). An inactive edge is called a lexical edge if it has a lexical category, otherwise it is a phrasal edge.

An active edge represents an incomplete constituent which needs some other complete constituents to make up a larger one. It is similarly characterized as above by *from*, *to*, *sub-inactive*, as well as *rule*, the referred grammar rule and *pos*, the position of the category in the rule that it is looking for.

A jump edge is a functional edge which links two different edges to indicate their link (described below) and to direct the parser to search through all the edges connected with each active edge during parsing. It is similarly characterized by *from* and *to*.

Meanwhile, the partial ordering relation on edges is defined as follows: For two edges  $E_i$  and

$E_j$ ,  $E_i \leq E_j$  iff  $\text{to}(E_i) \leq \text{from}(E_j)$ ; otherwise,  $E_i <> E_j$ . Also, the definition of edge connection is as follows: Edges  $E_i$  and  $E_j$  are said to be connected (i.e.  $\text{EConn}(E_i, E_j) = \text{true}$ ) iff  $E_i \leq E_j$  and either  $\text{to}(E_i) = \text{from}(E_j)$ , or there is a jump edge  $E_k$  so that  $\text{to}(E_i) = \text{from}(E_k)$  and  $\text{to}(E_k) = \text{from}(E_j)$ . In other words,  $E_i$  and  $E_j$  are connected only when the end vertex of  $E_i$  is the begin vertex of  $E_j$ , or when there is a jump edge linking them together. For example, in the chart given in Figure 7 for the sample word lattice given in Figure 3 (the details will be explained in the next section),  $\text{EConn}(E_3, E_6) = \text{true}$  due to the existence of Jump3 linking  $E_3$  and  $E_6$ , but  $\text{EConn}(E_1, E_6) = \text{false}$  due to  $E_3$  and  $E_4$  falling in between. This jump edge and the new link is the primary difference between the conventional chart and our augmented chart.

#### 4. The mapping from a word lattice to the augmented chart

Before parsing is performed, the input word lattice has to be represented by the augmented chart. We will describe such a mapping procedure below.

At the beginning of the mapping procedure, we first have to consider a situation in which additional word hypotheses must be inserted into the input lattice to avoid any important word being missed in the sentence. A good example for such a situation is in Figure 8, where the time segment for the word hypothesis  $w_i$  (the word "same") is from 10 to 20, and the one for  $w_j$  (the word "message") is from 14 to 30. Apparently all four cases are possible for this situation:  $w_i$  is a correct word but  $w_j$  is not,  $w_j$  is correct but  $w_i$  is not, both  $w_i$  and  $w_j$  are correct because they share a common phoneme (m) in the continuous co-articulated acoustic signal, or both  $w_i$  and  $w_j$  are not correct. A simple approach to be used here is for two additional word hypotheses  $w_{i1}$  (also "same", but from 10 to 17) and  $w_{j1}$  (also "message", but from 17 to 30) to be inserted into word lattice  $W$ , so that all the four possible cases will be properly considered during parsing and no words will be missed. This can be achieved by performing the following routine.

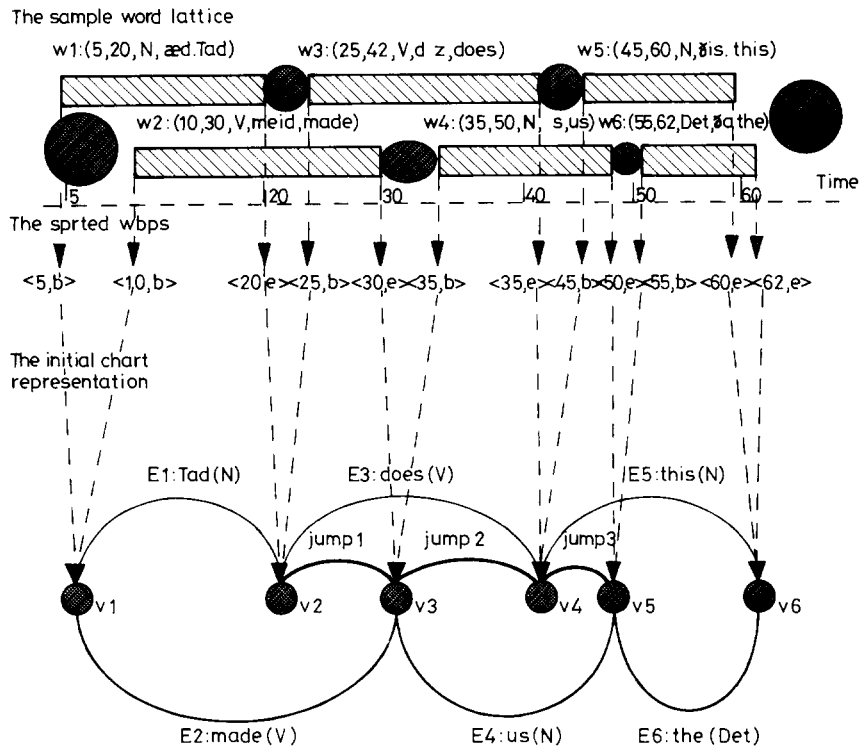


Fig. 7. The mapping results of the sample word lattice in Figure 3.

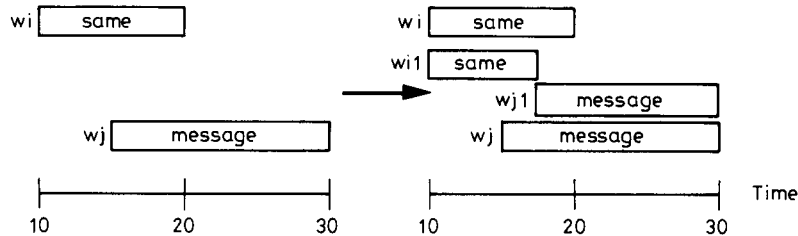


Fig. 8. The situation in which additional word hypotheses are inserted.

In the case of any pair of overlapping word hypotheses  $w_i$  and  $w_j$  in  $W$  where  $i < j$  if  $\text{phone}(w_i)$  and  $\text{phone}(w_j)$  share some common phonemes at the overlapping ends, two boundary aligned word hypotheses  $w_{i1}$  and  $w_{j1}$  are inserted into  $W$  so that  $w_{i1}$  contains exactly the same information as  $w_i$  except  $\text{end}(w_{i1}) = [\text{begin}(w_j) + \text{end}(w_i)]/2$ , and  $w_{j1}$  contains exactly the same information as  $w_j$  except  $\text{begin}(w_{j1}) = [\text{begin}(w_j) + \text{end}(w_i)]/2$ .

The boundary points (either beginning or ending) of any word hypothesis should now corre-

spond to a vertex in the chart. To start the mapping procedure, all the boundary points of the word hypotheses of  $W$  have to be sorted into an ordered sequence; the order of the sequence is defined as follows. To any pair of word boundary points (wbp)  $x$  and  $y$ , if  $x$  and  $y$  are distinct then their order is based on order in time; if  $x$  and  $y$  are identical then the beginning wbp is after the ending wbp. For convenience, let function  $\text{Order}(x)$  denote the sequence number of any wbp  $x$  in the sorted sequence.

Next, to each wbp  $x$ , the vertex assignment depends on its preceding wbp  $y$ ; i.e., the wbp  $y$  for which  $\text{Order}(y) = \text{Order}(x) - 1$ . As was discussed in Figure 9 in detail, for totally four possible cases of  $x$  and  $y$ , only for the first case in which  $x$  is an ending wbp but  $y$  a beginning wbp, two different vertices should be assigned to  $x$  and  $y$ . But in all other cases,  $x$  and  $y$  should be given the same vertex. Let function  $\text{Vertex}(x)$  denote this assignment. It is formally described below:

Suppose  $x$  and  $y$  are wbp's  
 if  $\text{Order}(x) = 1$  then  $\text{Vertex}(x) = 1$   
 else

let  $\text{Order}(y) = \text{Order}(x) - 1$ ,  
 if  $x$  is an ending wbp and  $y$  is a beginning wbp  
 then  $\text{Vertex}(x) = \text{Vertex}(y) + 1$ ;  
 else  $\text{Vertex}(x) = \text{Vertex}(y)$ ;

Now, for each word hypothesis  $w_i$ , a lexical edge can be constructed. The function  $\text{Edge}(w_i)$  for a word hypothesis  $w_i$  is then exactly specified by the two vertices assigned to the two wbp's on both ends of the word hypothesis, i.e. the definition of  $\text{Edge}(w_i)$  is  $\text{Edge}(w_i) = \langle \text{Vertex}(\text{begin}(w_i)), \text{Vertex}(\text{end}(w_i)) \rangle$ . Finally, if there is no complete lexical edge existing between any pair

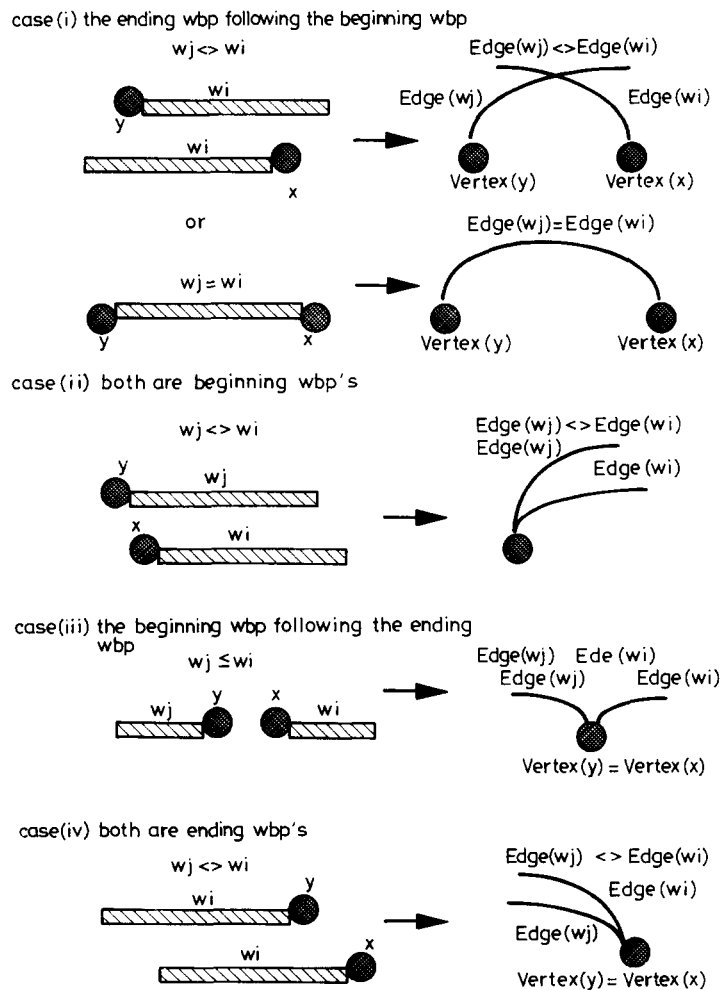


Fig. 9. Vertex assignment of the word boundary points. In the above totally four possible cases of a wbp  $x$  with its preceding wbp  $y$  are addressed, where  $x$  is a wbp of word hypothesis  $w_i$  and  $y$  is a wbp of  $w_j$ . In case (i),  $x$  is an ending wbp but  $y$  is a beginning. Apparently  $x$  and  $y$  should be assigned to different vertices such that the relation between  $w_i$  and  $w_j$  can be preserved. However, in other cases (ii), (iii) and (iv),  $x$  and  $y$  should be assigned to the same vertex instead.

of vertices  $v_i$  and  $v_j$ , a jump edge from  $v_i$  to  $v_j$  is constructed to link  $v_i$  and  $v_j$ .

Bearing the above definitions in mind, what follows is a five-step initialization procedure to map a word lattice to the augmented chart.

Procedure: Chart-Initialization

Input: a word lattice  $W = \{w_1, \dots, w_m\}$

Output: an initial augmented chart

Step 1: Call insert routine.

Step 2: According to the definition of the function  $\text{Order}(x)$ , sort all word boundary points (wbp's) for all word hypotheses in  $W$  into an ordered sequence.

Step 3: To each of these wbp's, based on the definition of the function  $\text{Vertex}(x)$ , assign a corresponding vertex in the chart to each wbp.

Step 4: to each word  $w_i$  in  $W$ , construct a lexical edge as follows:

for each word  $w_i$  in  $W$

let  $\langle f, t \rangle = \text{Edge}(w_i)$

if the vertex  $f$  hasn't been created yet  
then construct the vertex  $f$ ;

if the vertex  $t$  hasn't been created yet  
then construct the vertex  $t$ ;

construct a lexical edge  $E_i$  from vertex  $f$  to vertex  $t$ , where

$\text{from}(E_i) = f$ ,  $\text{to}(E_i) = t$ ,  $\text{name}(E_i) = \text{name}(w_i)$ ,  $\text{cat}(E_i) = \text{cat}(w_i)$ ,  $\text{sub-inactive}(E_i) = \emptyset$ .

end for

Step 5: Construct jump edges.

for each pair of vertices  $v_i$  and  $v_j$

if there are no complete lexical edges existing between  $v_i$  and  $v_j$

then construct a jump edge from  $v_i$  to  $v_j$ ;

end for

Using the above procedure, the mapping results of the sample word lattice in Figure 3 are shown in Figure 7, where the sample word lattice is at the top, the sorted wbp's (specified by a time scale and whether it is a beginning or ending wbp in Figure 7) are in the middle, and the resulting initial chart is at the bottom.

## 5. Some properties of the mapping relation

In this section, some properties of the mapping relation described in the previous section will be discussed.

**LEMMA 1.** *If  $x$  and  $y$  are wbp's of word hypotheses in  $W$ , then  $\text{Order}(x) \leq \text{Order}(y)$  iff  $\text{Vertex}(x) \leq \text{Vertex}(y)$ .*

**PROOF.** This can be easily proved by the definitions of the functions  $\text{Order}$  and  $\text{Vertex}$ .

**LEMMA 2.** *For any pair of word hypotheses  $w_i$  and  $w_j$  in  $W$ , if  $w_i \leq w_j$  then  $\text{Edge}(w_i) \leq \text{Edge}(w_j)$ .*

**PROOF.** According to the partial ordering definition of word hypotheses, if  $w_i \leq w_j$  then  $\text{end}(w_i) \leq \text{begin}(w_j)$ . According to Lemma 1,  $\text{Order}(\text{end}(w_i)) \leq \text{Order}(\text{begin}(w_j))$ , therefore  $\text{Vertex}(\text{end}(w_i)) \leq \text{Vertex}(\text{begin}(w_j))$ . Based on the partial ordering definition of edges,  $\text{to}(\text{Edge}(w_i)) = \text{Vertex}(\text{end}(w_i)) \leq \text{from}(\text{Edge}(w_j)) = \text{Vertex}(\text{begin}(w_j))$ , that is,  $\text{Edge}(w_i) \leq \text{Edge}(w_j)$ .

**LEMMA 3.** *For any pair of word hypotheses  $w_i$  and  $w_j$  in  $W$ , if  $w_i <> w_j$  then  $\text{Edge}(w_i) <> \text{Edge}(w_j)$ .*

**PROOF.** If  $w_i <> w_j$ , thus  $i < j$  and  $\text{begin}(w_j) < \text{end}(w_i)$  or  $j < i$  and  $\text{begin}(w_i) < \text{end}(w_j)$ . Assume  $j < i$  and  $\text{begin}(w_i) < \text{end}(w_j)$  without loss of generality, one thus has  $\text{Vertex}(\text{begin}(w_i)) < \text{Vertex}(\text{end}(w_j))$  and  $\text{from}(\text{Edge}(w_i)) < \text{to}(\text{Edge}(w_j))$ , which means  $\text{Edge}(w_i) <> \text{Edge}(w_j)$ .

**PROPERTY 1.** *After the mapping relation, the ordering relation among all word hypotheses in the word lattice is completely preserved in the corresponding edges in the augmented chart.*

**PROOF.** This can be proved with Lemmas 2 and 3.

**COLLORARY 1.** *For any pair of word hypotheses  $w_i$  and  $w_j$  in  $W$ ,  $\text{EConn}(\text{edge}(w_i), \text{Edge}(w_j)) = \text{true}$  iff  $\text{WConn}(w_i, w_j) = \text{true}$ .*



**PROPERTY 2.** For each sentence hypothesis  $S_i$  of  $W$ ,  $S_i = w_{i1}, \dots, w_{ir}$ , the corresponding edges for each pair of adjacent component word hypotheses are all connected, i.e.,  $\text{EConn}(\text{Edge}(w_{ik}), \text{Edge}(w_{ik+1})) = \text{true}$  for all  $1 \leq k \leq r - 1$ .

**PROOF.** According to the definition of sentence hypothesis, each pair of adjacent word hypotheses  $w_{ik}$  and  $w_{ik+1}$  is such that  $w_{ik} \leq w_{ik+1}$ .  $\text{WConn}(w_{ik}, w_{ik+1}) = \text{true}$ , so  $\text{EConn}(\text{Edge}(w_{ik}), \text{Edge}(w_{ik+1})) = \text{true}$  (with Corollary 1).

**PROPERTY 3.** In the case that all the word hypotheses in  $W$  do not overlap with one another in time, i.e., for each pair of  $w_i$  and  $w_{i+1}$  in  $W$ , such that  $\text{end}(w_i) \leq \text{begin}(w_{i+1})$  then  $\text{Vertex}(\text{end}(w_i)) = \text{Vertex}(\text{begin}(w_{i+1}))$ .

**PROOF.** Because all the word hypotheses in  $W$  do not overlap, the sorted wbp's will be in the following way: the beginning wbp of  $w_1$ , the ending wbp of  $w_1$ , the beginning wbp of  $w_2$ , the ending wbp of  $w_2$ , ..., the ending wbp of  $w_m$ . According to the definition of vertex assignment for a wbp, each wbp will be assigned a vertex and  $\text{Vertex}(\text{end}(w_i)) = \text{Vertex}(\text{begin}(w_{i+1}))$ .

Property 3 implies that when the input word lattice can be reduced to a single sequence of word hypotheses, its representation in the augmented chart can also be reduced to a normal representation in a conventional chart. Therefore parsing a single sequence of word hypotheses in this augmented chart parser will be exactly the same as parsing it in a conventional chart parser.

## 6. The augmented chart parsing

The fundamental principle of the augmented chart parsing is exactly the same as that in the conventional chart parsing. The only difference is the definition of edge connection, as discussed in the above. In the following, this primary difference will be described in detail.

The fundamental principle of chart parsing is: Whenever an active edge  $A$  is connected to an inactive edge  $I$  which satisfies  $A$ 's conditions for extensions, a new edge  $N$  covering both is built.

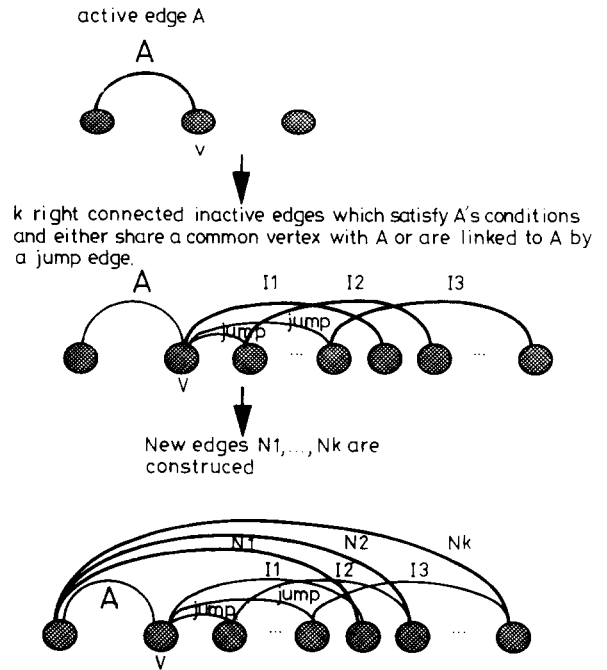


Fig. 10. The fundamental principle of the augmented chart parsing.

Now, in the augmented chart parsing this principle still holds; except that the inactive edge  $I$  does not have to share the same vertex with the active edge  $A$ ; instead, it can be separated from active edge  $A$ , but there must be a jump edge linking edges  $A$  and  $I$ . The diagram in Figure 10 shows this difference (in a left-to-right parsing example): First of all, active edge  $A$  was constructed, and  $k$  connected inactive edges  $I_1, \dots, I_k$  were found, each of these satisfies  $A$ 's conditions and either shares a common vertex with  $A$  or is linked to  $A$  by a single jump edge.  $k$  new edges  $N_1, \dots, N_k$  could then be built. Because these jump edges were constructed in the chart initialization, no difficulties arose in the search for the connected edges. The details will be described below.

When active edge  $A$  has been built, it starts to search for its connected inactive edges. If connected inactive edge  $I_i$  satisfies  $A$ 's conditions, i.e.  $\text{cat}(A) = \text{cat}(I_i)$  (in a lexicalized grammar different conditions may apply), then a new edge  $N_i$  will be constructed. There are two kinds of edge constructions in  $N_i$ :  $N_i$  may be a complete constituent, i.e., if all the necessary categories on the right-hand side of rule( $A$ ) have been found, then

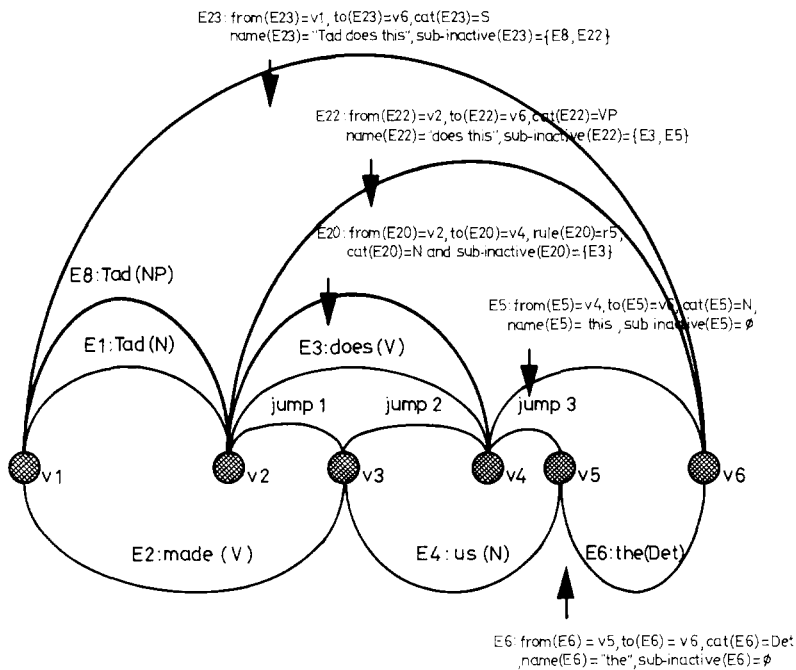


Fig. 11. Partial results of the parsing of the sample word lattice in Figure 3 using the augmented chart.

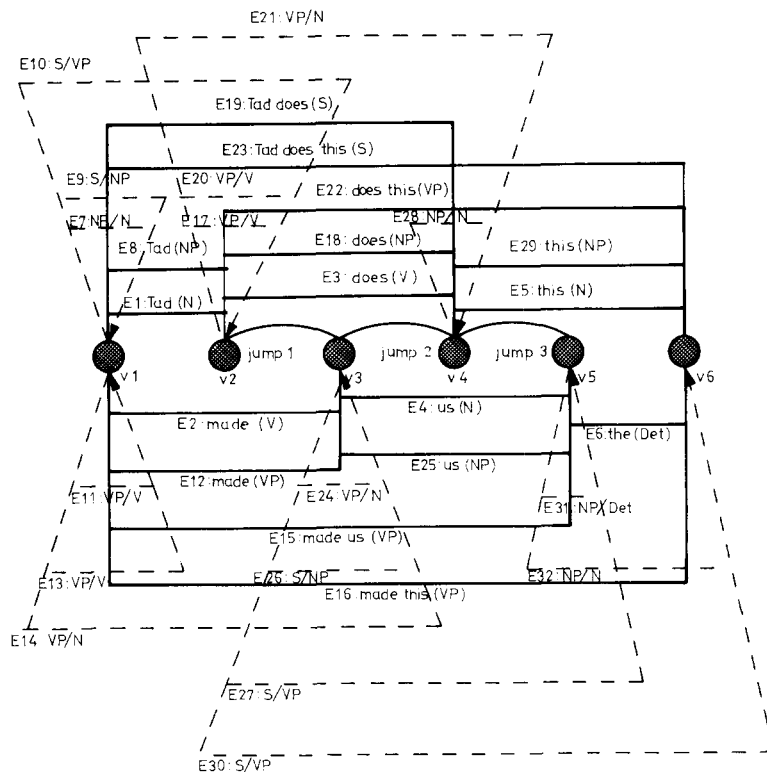


Fig. 12. The final results of parsing the sample word lattice in Figure 3 with the augmented chart parsing scheme. The solid line denote constructed inactive edges and the dashed lines active edges.

$N_i$  will be constructed with  $\text{from}(N_i) = \text{from}(A)$ ,  $\text{to}(N_i) = \text{to}(I_i)$ ,  $\text{cat}(N_i) = \text{the category of the left-hand side of rule}(A)$ , and  $\text{sub-inactive}(N_i) = \text{sub-inactive}(A) \cup \{I_i\}$ ; on the other hand,  $N_i$  may be an incomplete constituent, with  $\text{from}(N_i) = \text{from}(A)$ ,  $\text{to}(N_i) = \text{to}(I_i)$ ,  $\text{rule}(N_i) = \text{rule}(A)$ ,  $\text{cat}(N_i) = \text{the next necessary category in the right-hand side of rule}(A)$  (it can be found with  $(\text{pos}(A))$ ), and  $\text{sub-inactive}(N_i) = \text{sub-inactive}(A) \cup \{I_i\}$ . For example, in Figure 11 suppose the active edge  $E_{20}$  is newly constructed, where  $\text{from}(E_{20}) = v_2$ ,  $\text{to}(E_{20}) = v_4$ ,  $\text{rule}(E_{20}) = r_5$ ,  $\text{cat}(E_{20}) = N$  and  $\text{sub-inactive}(E_{20}) = \{E_3\}$ . Also, in the augmented chart lexical edges  $E_5$  and  $E_6$  are connected to inactive edge  $E_{20}$ , where  $E_5$  shares the same vertex  $v_4$  with  $E_{20}$  while  $E_6$  has a jump edge  $\text{jump}_3$  to link to  $E_{20}$ . Due to  $\text{cat}(E_5) = \text{cat}(E_{20})$  but  $\text{cat}(E_6) \neq \text{cat}(E_{20})$ ,  $E_5$  is the only satisfactory candidate edge. Therefore only the inactive edge  $E_{22}$  is constructed with  $\text{from}(E_{22}) = v_2$ ,  $\text{to}(E_{22}) = v_6$ ,  $\text{cat}(E_{22}) = \text{VP}$ ,  $\text{sub-inactive}(E_{22}) = \{E_3, E_5\}$ . The final results of the parsing of the sample word lattice are given in Figure 12.

## 7. The parsing structures

In general, the parsing structure to be assigned to an inactive edge with a rule-based grammar is a syntactic parsing tree associated with the edge. In our augmented chart, this kind of parsing structure can also be easily obtained, just as in a conventional chart.

Let  $E_i$  be an inactive edge. The parsing structure that  $E_i$  represents is described as follows. Let  $\text{cat}(E_i)$  denote the syntactic category of  $E_i$ . If  $E_i$  is a lexical edge, then the parsing structure that  $E_i$  represents is simply itself; if  $E_i$  is a phrasal edge, then the edges listed in  $\text{sub-inactive}(E_i)$  denote the immediate dominant components of  $E_i$ , and the ordering of these edges denotes the precedence. At the same time, the parsing structures of these components are defined recursively in a similar manner.

For example, Figure 13 is the corresponding parsing structure of inactive edge  $E_{23}$  in Figure 11. Another kind of parsing structure for use in a lexicalized grammar will also be introduced later in Section 9.

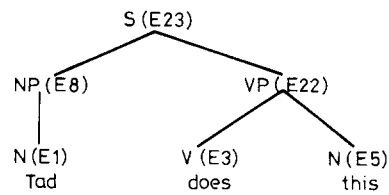


Fig. 13. A sample parsing structure.

## 8. Preliminary experimental results

In order to see how the above concept for augmented chart parsing works, a bottom-up and left-to-right parser based on the proposed augmented chart (also capable of performing conventional chart parsing) has been implemented and tested in some preliminary experiments. A large set of Chinese word lattices obtained by an acoustic signal processor which recognizes Mandarin speech was used as the input to the parser. Due both to the large number of homonyms existing in Chinese and to the uncertainty and error in speech recognition, there was a very high degree of lexical ambiguity in the input lattices. The phrase structure grammar rules used for the Chinese sentences in this test are listed in Figure 14. One example of such a Chinese word lattice is in Figure 15. The results show that every possible constituent for the input word lattice can be constructed, and no constituent needs to be built more than once using augmented chart parsing. In order to compare the efficiencies that an augmented chart parser can provide, an edge reduction ratio is defined as follows:

$$\text{edge reduction ratio} = (\text{total number of edges built in the augmented chart parsing}) / (\text{total number of edges built in conventional chart parsing})$$

to indicate the reduction of the total number of edges which have to be constructed in the augmented chart as compared to the conventional chart, assuming that all possible sentence hypotheses have to be parsed exhaustively. According to the experimental results, this edge reduction ratio is of the order of  $1/30 \sim 1/80$  for our input Chinese word lattices. Apparently this ratio depends heavily on the degree of ambiguity of the input word lattices. The computational complexity can always be reduced significantly, however.

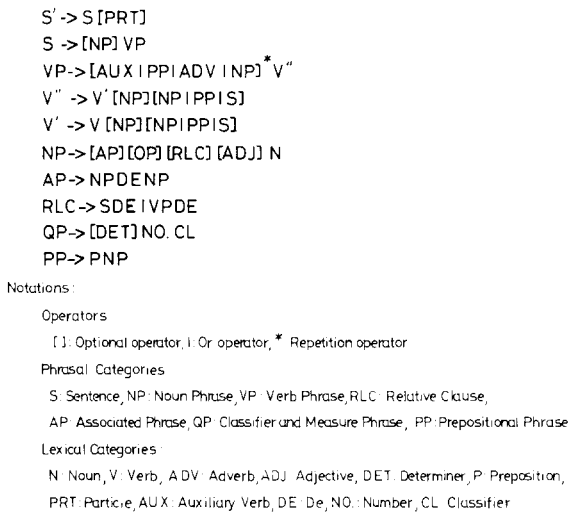


Fig. 14. The phrase structure rules used in the tests.

Some experiments on English word lattices have also been performed (a simplified parsing algorithm is listed in the appendix). For the sample word lattice in Figure 3, a total of 32 edges were constructed and the final results are given in Figure 12 as previously discussed. All the 32 edges are listed in Figure 16, where columns 1 and 2 list the edge numbers and names, while columns 3 to 7 record the edge construction operations in parsing each of the five sentence hypotheses. Every "X" in Figure 16 indicates that the corresponding edge was constructed during the parsing of the corresponding sentence hypothesis. The last column is the total number of repetitions in the construction of each edge when all the sentence hypotheses were parsed in a conventional chart. The last row shows that for this example, a total of 70 edges have to be constructed in conventional

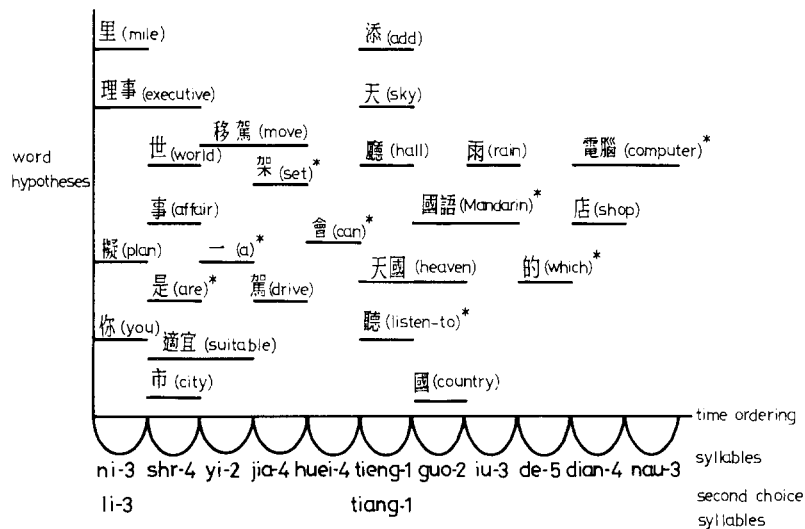


Fig. 15. The example given here is taken from the Mandarin Chinese sentence: ni-3 "you" shr-4 "are" yi-2 "a" jia-4 "set" hui-4 "can" tieng-1 "listen to" guo-2 iu-3 "Mandarin" de-5 "which" dian-4 nan-3 "computer" (you are a computer which can listen to Mandarin, 你是一架會聽國語的電腦), where the syllables are represented in Mandarin Phonetic Symbols II (MPS-II), with the integers (1 to 5) indicating the tone. The possible word hypotheses are shown above, where the horizontal axis denotes the time ordering of the syllables and the vertical scale shows the corresponding word hypotheses, of which only those denoted by "\*" are correct words. In this example all the syllables were clearly identified and correctly recognized, except that two syllables (the first syllable ni-3 and the sixth syllable tieng-1) were confused by a second candidate (li-3 and tiang-1, respectively). This ambiguity was therefore primarily due to the large number of homonyms in Chinese. The line segments under each word hypothesis indicate whether the word hypothesis is composed of one or two syllables. In our analysis, as many as 470 sentence hypotheses were obtained from this example of a word lattice, with most of the syllables immediately recognized, and the experimental results show that for this example, a total of 58132 edges have to be built in conventional chart parsing, while only 925 edges are necessary in the augmented chart parsing. The edge reduction ratio for this example is 1/62.8.

Edge No	Edge name	Tad does this	Tad does the	Tad us the	made us the	made this	Number of repetition
1	Tad (N)	X	X	X			3
2	made (V)				X	X	2
3	does (V)	X	X		X		3
4	us (N)			X			1
5	this (N)	X				X	2
6	the (Det)		X	X	X		3
7	NP/N	X	X	X			3
8	Tad (NP)	X	X	X			3
9	S/NP	X	X	X			3
10	S/VP	X	X	X			3
11	VP/V				X	X	2
12	made (VP)				X	X	2
13	VP/V				X	X	2
14	VP/N				X	X	2
15	made us (VP)				X		1
16	made this (VP)					X	1
17	VP/V	X	X				2
18	does (NP)	X	X				2
19	Tad does (S)	X	X				2
20	VP/V	X	X				2
21	VP/N	X	X				2
22	does this (VP)	X					1
23	Tad does this (S)	X					1
24	VP/N			X	X		2
25	us (NP)			X	X		2
26	S/NP	X		X	X	X	4
27	S/VP			X	X		2
28	NP/N	X				X	2
29	this (NP)	X				X	2
30	S/VP	X				X	2
31	NP/Det		X	X	X		3
32	NP/N		X	X	X		3
Sum		18	14	13	14	11	70

Fig. 16. A table listing the construction of each edge while parsing the sample word lattice in Figure 3.

chart parsing, while only 32 are necessary in augmented chart parsing. According to the above formula, the edge reduction ratio for this example is  $1/2.2$  for this word lattice, with a relatively low degree of ambiguity. Moreover, it can be seen that all these 32 edges are necessary (because no entry in the last column is zero), but no edge has to be built more than once when the proposed augmented chart is used.

## 9. Further extensions

In general, there are three basic issues in parser design, i.e., the structure to be assigned, the type of grammar and the recognition algorithm. Many parsers very often employ phrase structure descriptions, rule-based grammars and derivation or transition oriented recognition. Whereas the augmented chart parsing scheme proposed here has been shown to be very useful and efficient in these areas, we now propose to

demonstrate that the proposed augmented chart is equally useful and efficient in other fields such as lexicalized grammar, e.g. HPSG (Pollard and Sag, 1987). A lexicalized grammar is a grammar whose syntactical relationships are stated as part of the lexical description. In such cases, the structures to be assigned to the input in the augmented chart may be extended to attribute-value matrices (complex feature structures) instead of syntactic parsing trees, and the recognition algorithm may rely on the head-driven slot and filler principle instead of derivation oriented recognition. In other words, such an extension is in fact straightforward. The categories of edges (inactive or active) are now replaced by feature structures, and information such as *rule* and *pos* becomes unnecessary. The fundamental principles of augmented chart parsing remains unchanged, but with a slightly different interpretation. Active edge *A* should now be a head constituent (e.g. the head noun in an NP or the head verb in a VP) with some empty slots, and inactive edge *I* is a necessary constituent of *A* that can fill some of the slots. An example in Figure 17 shows this extension, where *A* is a head verb constituent with an unknown slot "object", and *I* is an NP constituent that can fill the empty slot, so constructing a complete VP constituent *N*.

Moreover, in order to increase the flexibility of the slot and filler principle further, some approaches such as island parsing (Stock et al., 1988) and discontinuous segmented parsing (Hellwing, 1988) have been developed. The augmented chart proposed here can also be easily extended to these cases. In other words, the pres-

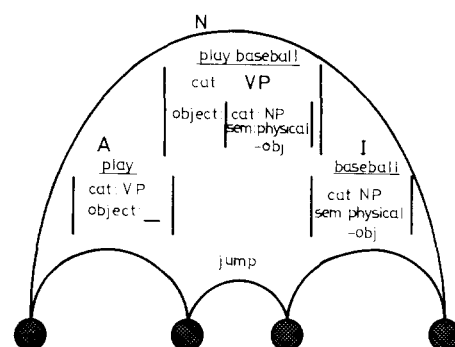


Fig. 17. An extension to a lexicalized grammar.

ent augmented chart parsing scheme is a general approach on the basis of which various grammar formalisms and parsing strategies can be implemented in different speech recognition applications. For example, some approaches may utilize phrase structure grammars and left-to-right parsing strategies to treat the word lattices without missing words; while some other approaches may use lexicalized grammars and island-driven parsing strategies to recover missing words from seriously corrupted word lattices. In comparison with Tomita's approach (Tomita, 1986), which utilizes an LR parsing table to perform left-to-right predictive parsing on word lattices, the present parsing scheme basically possesses more flexibility for further extension. In fact, Tomita's approach is a specialized parsing algorithm and is therefore inherently limited in some applications where phrase structure grammars and/or left-to-right parsing strategies are insufficient. This applies, for instance, in the radical lexicalized analysis for languages with very free word order (Karttunen, 1986), where the lexical entries of words encode virtually all the information about how the words are combined into phrases; there is thus no separate syntactic component. However, the present parsing scheme can easily be extended to these applications as described above.

## **10. Discussions of an example of application**

Because the augmented chart and word lattice parsing scheme proposed here is a general approach, it is flexible enough to be used in various applications for different problems. An example of a robust word lattice parsing system for handling seriously corrupted word lattices will now be briefly summarized and discussed. Since word lattices produced from an acoustic signal processor may be seriously corrupted, the parsing scheme for handling such seriously corrupted word lattices needs to be able to tolerate, or even to compensate for, recognition errors occurring in the acoustic signal processor. Nevertheless, for most sentence parsers, grammatical information is the only knowledge adopted. In dealing with seriously corrupted word lattices, there are at

least two problems with which this sort of parser has difficulty. The first problem is that very often more than one grammatically valid sentence hypothesis may be produced due to a high degree of lexical ambiguity, and there is no evident strategy based on grammatical knowledge alone which will be sufficient to select the most promising result. The second problem is that some keywords may be missed in the word lattices, with the result that no grammatically valid sentence hypothesis can be identified. To cope with both problems, some unified methods integrating both the grammatical and the statistical approaches (Derouault and Merialdo, 1986; Chien et al., 1990; Chien, 1991) seem to be useful, since the features of these approaches are basically complementary. While the statistical approaches predict very well locally, and can even tolerate some errors of recognition, the grammatical ones are better at checking the global structures, finding out the exact meaning or sentence structure, and can sometimes compensate for recognition errors. As is described below, such an example word lattice parsing system based on the augmented chart discussed here (Chien et al., 1990; Chien, 1991) has been successfully implemented.

In this example, an efficient parsing algorithm based on the present augmented chart parsing scheme for Chinese word lattices has been developed which integrates the unification grammar (a grammatical approach) and a Markov language model (a statistical approach) in order to handle the difficult problems mentioned above. The parsing algorithm is island-driven and best-first. That is, only grammatically unifiable constituents (islands) can be grouped as a new constituent (island-driven), and at any given moment only the constituent with the highest probability (obtained from the Markov language model) can be really constructed (best-first). In this way, not only can the interference from noisy word hypotheses and the high computational complexity due to the huge search space be significantly reduced, but the proper selection of parsing strategies can produce reasonable results even if the input lattices are seriously corrupted. For instance, in order to deal with the first problem discussed above, one can select the sentence hypothesis with the highest probability as the result; or to save computa-

tion, one can select the first sentence constituent obtained during parsing with a satisfactory probability as the result. On the other hand, to handle the second problem discussed above, one can select a sequence of connected constituents with the highest probability as a partial result, or one can even feed such information back into the acoustic signal processor to ask for more word hypotheses at the time segments where some keywords may be missed. This is, in fact, significantly different from those methods based on Tomita's algorithm (Tomita, 1986) which simply rely on the guidance of a syntactically based parsing table and a strictly left-to-right parsing strategy and are therefore relatively inadequate to handle the problem of seriously corrupted word lattices.

## 11. Concluding remarks

In this paper, an augmented chart data structure for word lattice parsing is proposed. It is able to represent and parse a lattice of words very efficiently without changing the fundamental principles of chart parsing. With augmented chart parsing, all possible constituents of an input word lattice can be constructed and no constituent needs to be built more than once. This will significantly reduce the computational complexity especially when serious lexical ambiguity exists in the input word lattice. This augmentation does not affect the original principles, operations and applications of the conventional chart parser. It is a general parsing scheme, independent of grammar formalisms and parsing strategies, and can thus be easily extended to different applications. This mented chart parsing scheme is therefore a very useful and efficient approach for speech recognition applications with a high degree of lexical ambiguity.

## References

- L.F. Chien (1991), Several efficient approaches for language modeling and processing in speech recognition applications, *PhD. Dissertation*, National Taiwan University, Taiwan, ROC.
- L.F. Chien, K.J. Chen and L.S. Lee (1990), "An augmented chart parsing algorithm integrating unification grammar and Markov language model for continuous speech recognition", *Proc. 1990 Internat. Conf. Acoust. Speech Signal Process.*, Albuquerque, NM, USA.
- Yen-Lu Chow and Salim Ronkos (1989), "Speech understanding using a unification grammar", *Proc. 1989 Internat. Conf. Acoust. Speech Signal Process.*, Glasgow, Scotland, pp. 727-730.
- A. Derouault and B. Merialdo (1986), "Natural language modeling for phoneme-to-text transcription", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-8, No. 6, pp. 742-749.
- P.J. Hayes et al. (1986), "Parsing spoken language: A semantic caseframe approach", *Proc. 11th Internat. Conf. Computational Linguistics*, University of Bonn, Germany, pp. 587-592.
- P. Hellwing (1988), "Chart parsing according to the slot and filler principle", *Proc. 12th Internat. Conf. Computational Linguistics*, Budapest, pp. 242-244.
- Chu-Ren Huang and Yu-Ling Shiu (1988), "Unification-based analysis and parsing strategy of mandarin particle question", *Proc. 1988 Internat. Computer Symposium*, Taipei, pp. 38-43.
- M. Kay (1980), "Algorithm schemata and data structures in syntactic processing", *Xerox Report*, CSL-80-12, Palo Alto.
- L. Karttunen (1986), Radical lexicalism, *CSLI Lecture Notes*, No. CSLI-86-68, Stanford University, Stanford, CA, USA.
- L.S. Lee and C.Y. Tseng (1987), "Mandarin speech input/output techniques for Chinese computer - The state of the art", *Proc. National Science Council (Physical Science and Engineering)*, No. 4, pp. 273-290.
- L.S. Lee et al. (1987), "The preliminary results of a Mandarin dictation machine based upon Chinese natural language analysis", *Proc. 1987 Internat. Joint Conf. Artificial Intelligence*, Milan, pp. 619-621.
- C. Pollard and I.A. Sag (1987), Information-based syntax and semantics, Vol. 1. Fundamentals, *CSLI Lecture Notes*, No. 12, Stanford University, Stanford, CA, USA.
- O. Stock et al. (1988), "Island parsing and bidirectional charts", *Proc. 12th Internat. Conf. Computational Linguistics*, Budapest, pp. 636-641.
- H. Thompson (1981), "Chart parsing and rule schemata in GPSG", *Proc. 19th Annual Meeting of the Association for Computational Linguistics*, Stanford University, Stanford, USA.
- M. Tomita (1986), "An efficient word lattice parsing algorithm for continuous speech recognition", *Proc. 1986 Internat. Conf. Acoust. Speech Signal Process.*, pp. 1569-1572.
- W.H. Ward et al. (1988), "Parsing spoken phrases despite missing words", *Proc. 1988 Internat. Conf. Acoust. Speech Signal Process.*, pp. 275-278.

## Appendix A. The bottom-up and left-to-right parsing algorithm

Procedure Parser( $W, G$ )

Input: (1) A word lattice  $W = \{w_1, \dots, w_n\}$

(2) A grammar,  $G$

Goal: According to the description of the grammar,  $G$ , to find all the grammatically valid sentence hypotheses and their corresponding sentence structures which can be constructed from  $W$ .

Global data structures:

(1) Chart ( $V, E$ )

(2) Agenda: The agenda is an LIFO list of expectations.

(An expectation is a pair of information  $\langle a, i \rangle$ , to denote the possible combination of active edge  $a$  and inactive edge  $i$ .)

Step 1: Call procedure Chart-Initialization( $W$ ), assuming  $w_{n+1}, \dots, w_{n+m}$  are newly inserted word hypotheses (This can be found in Section 4)

Step 2: For each lexical inactive edge  $E_i$  constructed in step 1,  $i = 1, \dots, n + m$

call procedure Expectation-Formation( $E_i$ );

pop an expectation  $\langle a, i \rangle$  from the agenda and then call procedure Edge-Construction( $a, i$ ) until the agenda is empty.

Procedure Expectation-Formation( $E_i$ )

Step 1: To each possible active edge  $E_a$  of  $E_i$ , where  $E_a$  is near the left of  $E_i$ ,  $\text{cat}(E_i) = \text{cat}(E_a)$  and either shares the same vertex with  $E_i$  or has a jump edge linking it to  $E_i$ , then add expectation  $\langle a, i \rangle$  to the agenda.

Step 2: Suppose  $\text{cat}(E_i) = A$  and  $\text{from}(E_i) = v$ , for all the rules of the form  $B \rightarrow r_1 \dots r_s$  in grammar  $G$  so that  $r_1 = A$ , introduce a new active edge  $E_n$  from  $v$  to  $v$  (provided that there is no such edge currently in the chart), where  $\text{from}(E_n) = v$ ,  $\text{to}(E_n) = v$ ,

$\text{pos}(E_n) = 1$ ,  $\text{rule}(E_n) = \text{the referred rule}$  and  $\text{sub-inactive}(E_n) = \emptyset$ . Then, to each  $E_n$  add expectation  $\langle n, i \rangle$  to agenda.

Procedure Edge-Construction( $a, i$ )

/\*  $E_a$  and  $E_i$  are connected edges, where  $E_a$  is active,  $E_i$  is inactive and  $\text{cat}(E_i) = \text{cat}(E_a)$  \*/

Step 1: If  $\text{rule}(E_a)$  is satisfied, then construct a new inactive edge  $E_n$ , where

$\text{from}(E_n) = \text{from}(E_a)$ ,  $\text{to}(E_n) = \text{to}(E_i)$ ,  $\text{cat}(E_n) = \text{the left-hand side of rule}(E_a)$  and  $\text{sub-inactive}(E_n) = \text{sub-inactive}(E_a) \cup \{i\}$ ; and then call procedure Expectation-Formation( $E_i$ )

Step 2: If there are other necessary categories in  $\text{rule}(E_a)$  then construct a new active edge  $E_n$ , where  $\text{from}(E_n) = \text{from}(E_a)$ ,  $\text{to}(E_n) = \text{to}(E_i)$ ,  $\text{rule}(E_n) = \text{rule}(E_a)$ ,  $\text{pos}(E_n) = \text{pos}(E_n) + 1$  and  $\text{sub-inactive}(E_n) = \text{sub-inactive}(E_a) \cup \{a\}$ .