Online Appendix to *An Empirical Approach to Temporal Reference Resolution*

Janyce M. Wiebe, Thomas P. O'Hara, Thorsten Öhrström-Sandgren & Kenneth J. McKeever

1. Temporal Reference Resolution Rules

1.1 Introduction

This document provides a detailed algorithm for temporal reference resolution as used in the Artwork system developed at New Mexico State Univsersity. The remainder of this section covers conventions used in the algorithm. The algorithm itself is in Section 1.2.

1.1.1 Abbreviations

1. **SpanUtt** = Spanish Utterance; a list of the Spanish words occurring in a sentence. For instance: "cinco de marzo no" would be represented as [cinco,de,marzo,no].
2. **TU** = Temporal Unit.
3. **ILT** = A structured representation of the meaning of an **SpanUtt**, possibly containing a TU. The ILT representation is defined by the Enthusiast project (Levin et al. 1995).
4. **DE** = Discourse Entity; Composed of a **SpanUtt** and an **ILT**.
5. **FL** = Focus List (ordered list of **DE**'s).
6. **RF** = Reference Frame; a Temporal Unit used as a temporal frame of reference.
7. **CF** = Certainty Factor.
8. **timeValue** = One of {**day, week, weekend, month, year**}.

1.1.2 Functions

1. **retrieveField**($FieldName$, *Structure*): returns the filler for *FieldName* in *Structure*, if *Structure* contains a $FieldName$, otherwise NULL.
2. **retrieveUtterance**(DE): returns the Spanish utterance from *DE*, if one exists, otherwise NULL.
3. **dateCopy**(TU): Copies the start dates of a TU to their corresponding end dates. The function returns the modified TU and a CF, as a tuple. The CF is based on the combination of fields copied.
4. **distanceFactor**(DE): Returns a number reflecting how far back on the focus list DE is.
5. **futureRange**(N_ILT): Returns a range in the future consistent with the ILT. For instance, "second week in August", interpreted as the second Monday through the following Friday in August, closest in the future. If successful, a new TU is returned, otherwise, NULL.

6. **last**($timeValue$, RF): returns the last *timeValue* before reference frame RF.

   Ex: last(**week**, [Tues 20 Aug 1996]
   Mon-Fri, 12-16th, August, 1996

7. **lastFLRange**($timeValue$, N): N is either '1', or '2'. If N = '1' then **lastFLRange** returns the TU on the focus list that most closely matches *timeValue* from the focus list. If N = '2', one matching TU on the list is skipped.

8. **merge**(TU_1, TU_2): if there exist conflicting fields in TU_1 and TU_2, return NULL. Otherwise return a TU which contains the union of all fields of TU_1 and TU_2.

9. **mergeUpper**(TU_1, TU_2): the same as **merge**, except only fields of the same or less specific levels than the most specific field in TU_1 are considered.

10. **mostSpecific**(X, TU): X is either **start**, **end**, or **both** to indicate the starting time, end time, or both the starting and end time, respectively. The function returns the specificity level of the most specific field of TU, where *month* is level 1. Ex: mostSpecific(start, TU) returns the specificity level of the most specific field in the start time of TU, and mostSpecific(*both*, TU) returns the specificity level of the most specific field of the entire TU.

11. **next**($timeValue$, RF): returns the next *timeValue* that follows the reference frame RF.

12. **nextInclToday**($timeValue$, RF): Same as **next**, but this version considers "today" as a possibility.

13. **resolveDeictic**(TU, $todaysDate$): resolves the deictic term TU with respect to the dialog date.

14. **applyRule**(TU, $RuleName$ [, *subcase*]): Invokes rule *RuleName*. Returns the return value of *RuleName*.

15. **this**($timeValue$): Returns the current $timeValue$ with respect to the conversation date.

    Ex: Dialog date is Thursday, 22th, August, 1996
    "This week has been long.",
    Returns Mon-Fri, 19th-23rd, August, 1996

16. **isDeictic**(TU): returns TRUE if TU contains deictic information, otherwise NULL.

17. **isRange**(X, TU): returns TRUE, if TU is an X, where X = {**week, day, time**}, otherwise returns NULL.

18. **moreEquSpecific**(TU_1, TU_2): returns TRUE if TU_1 is either more specific than TU_2, or if the two have the same level of specificity, otherwise, NULL.

19. **moreSpecificLow**(TU_1, TU_2): same as **moreEqu Specific**, but only tests levels of less or equal specificity as time_of_day.

1.1.3 Typographic Conventions

Variables are *italicized* (e.g., *TU, startTimes*). Variable values (e.g., **month, date**), keywords (e.g., **if, then, else**), and conceptual notations are in **bold-face**. Function names are in roman (e.g., merge(TU_1, TU_2)). **X**$\rightarrow$**Y** refers to the field(s) Y within X (e.g., *TU*$\rightarrow$*startTimes, TU*$\rightarrow$*endTimes, TU*$\rightarrow$*name*) % Denotes comments in rules. **forward-looking-adjective** are adjectives that indicate a time in the future. For instance, "next", "following", or in Spanish "proximo", "siguiente", "viene".

1.2 Rules

Rules are invoked in two different ways: one way for rules that do not access the focus list, and one for those that do. For rules that do not access the focus list the procedure is as follows.

1. Extract all of the *TU*s in the *DE*.
2. Apply the rule to each *TU*.
3. Put the resulting *TU*s back in the same order as they appeared in the original *TUL*.

For rules the do access the focus list, a similar process is performed, but with one important difference: When accessing the specific rule for the first Temporal Unit, which entity off the focus list that was used is recorded, and the remaining temporal units are forced to use the same one.

1.2.1 Rules for deictic relations

Deictic Rule 1: *(A type of relation* **D-simple***)*
A deictic expression is resolved into a time interpreted with respect to the dialog date (e.g., "Tomorrow", "last week").

if isDeictic(*TU*) **then**
  return $\langle$0.9, merge(*TU*, resolveDeictic(*TU*, todaysDate))$\rangle$
else fail
% *Ex: "Let's meet tomorrow", with an interpretation of Mon 24 Sept*

Deictic Rule 2: *(A type of relation* **D-frame-of-reference***)*
A forward time is calculated with respect to today as a frame of reference.

Subcase i:
  if not isDeictic(*TU*) **then**
    SpanUtt = retrieveUtterance(ILT)
    if forward-looking-adjective $\cap$ *SpanUtt* $\neq \{\}$ **then**
      if(mostSpecific(**start**, *TU*) $\leq$ **date**) **then**
        new_*TU* = next(mostSpecific(start,*TU*), todaysDate)
        return$\langle$0.2, merge(*TU*, *new_TU*)$\rangle$
      else fail

else fail

else fail

% Ex: "How about if we meet next Monday?", interpreted as Monday, Sep 30

Subcase ii:

if not isDeictic(TU) **then**

if(mostSpecific(**start**, TU) $\leq$ **date**) **then**

new_TU = nextInclToday(mostSpecific(start, TU), todaysDate)

return$\langle$0.3, merge(TU,new_TU)$\rangle$

else fail

else fail

% Ex: "Hmmm, how about Monday?" interpreted as the current *Monday, that is Mon 23 Sept. This rule differers from the previous one only in that the current date is also taken into consideration.*

The following cover special cases not discussed in the body of the paper.

Deictic Rule 3: *(A type of relation* **D-simple***)*

if not isDeictic(TU) **then**

if(**the_rest_of** $\in$ $TU \rightarrow Spec$) **then**

$newTU$ = mergeAll(the_rest_of($Name$)

% Ex: "I can meet Tuesday", resolved to Tue 25 Sept

"Good, I can meet for the rest of the week as well",

resolved to Wed 26 Sept - Fri 28 Sept"

else if(**the_end_of** $\in$ $Spec$) $\wedge$ (**this** $\in$ $Spec$) **then**

$newTU$ = the_end_of($Name$)

% Ex: "I can meet Monday", resolved to Mon 24 Sept

"No – better at the end of the week.",

resolved to Thu 27 Sept - Fri 28 Sept"

else if(**the_end_of** $\in$ $Spec$) $\wedge$ (**next** $\in$ $Spec$) **then**

$newTU$ = the_end_of_next($Name$)

% Ex: "I can meet at the end of next month", resolved to Mon 21 Oct - Thu 31 Oct

(The last ten days of the coming month)

else if(**last** $\in$ $Spec$) **then**

$newTU$ = last($Name$, TU)

% Ex: "Last Thursday's meeting was very productive.", resolved to Thu 17 Sept

else if(**this** $\in$ $Spec$) $\wedge$ (**coming** $\in$ $Spec$) **then**

$newTU$ = next($Name$, TU)

% Ex: "This coming Thursday is good.", resolved to Thu 27 Sept

else if(**this** $\in$ $Spec$) **then**

$newTU$ = this($Name$, TU)

% Ex: "This has been a good month.", resolved to Sat 1 Sept - Sun 30 Sept

else

$newTU$ = futureRange(TU)

if($newTU$) = **null then fail**
% Ex: "I can meet second week in October",
under the interpretation Mon 8 Oct to Fri 12 Oct.
return $\langle 0.5$, merge(TU, $newTU$)$\rangle$

Deictic Rule 4: *(A special case for handling end times) The mentioned time is not an interval; resulting frame's end dates correspond to the starting dates.*

if (simpleTime(TU) $\wedge$ {**sday, sdate, smonth**} is not **null**) **then**
$\langle newTU,\ CF\rangle$ = dateCopy(TU)
return $\langle CF$, merge(TU, $newTU$)$\rangle$
else fail
% Ex: "Let's meet Friday the 27 of September", under the interpretation that the start time and the end time both refer to Fri 27 Sept.

Deictic Rule 5: *(A type of relation* **D-simple***)*

if at least one of {$TU \rightarrow SMIN$, $TU \rightarrow SHOUR$, $TU \rightarrow STIME\_DAY$} $\neq$ **null** $\wedge$ current utterance does not contain a date **then**
% Make sure that no FL entry refers to **time_of_day** or lower
if ($\forall$ TU $\in FL$: mostSpecific(**both**, TU_{fl}) $\geq$ **time_of_day**) **then**
$today\_TU$ = today's date
return $\langle 0.5$, merge(TU, $today\_TU$)$\rangle$
else fail
else fail
% Ex: "How about 3:00?" resolved to 3:00 Mon 23 Sept

Deictic Rule 6: *(A type of relation* **D-simple***)*

if($TU \rightarrow Name$ = {**weekend**}) $\wedge$ ($TU \rightarrow Spec \in$ {**next, this, last**}) **then**
if($TU \rightarrow Spec$ = **next then**
new_TU = next(**weekend**)
if($TU \rightarrow Spec$ = **this then**
new_TU = this(**weekend**)
if($TU \rightarrow Spec$ = **last then**
new_TU = last(**weekend**)
return $\langle 0.8$, merge(TU, new_TU)$\rangle$
else fail
else fail

% Ex: "Is next weekend OK?" resolved to Sat 29-Sun 30 Sept

1.2.2 Rules for anaphoric relations

Anaphoric Rule 1: *(A type of relation* **A-co-reference***)*
The times discussed are similar; the resulting frame is the union of both times.

for each non-empty Temporal Unit TU_{fl} from FL, starting with most recent
  if (moreEquSpecific(TU, TU_{fl})) $\wedge$ (merge(TU, TU_{fl}) $\neq$ **null**) **then**
    return $\langle$0.8 - distanceFactor(TU_{fl}), merge(TU, TU_{fl})$\rangle$
  else fail
% *Ex:* *"Let's meet Tuesday." resolved to Tue 24 Sept*
"How about 2?" resolved to 2pm Tue 24 Sept

Anaphoric Rule 2: *(A type of relation* **A-less-specific***)*
The current utterance evokes a time that includes the time evoked by a previous time, and the current time is less specific.

for each non-empty Temporal Unit TU_{fl} from FL, starting with most recent
  if not (moreEquSpecific(TU, TU_{fl})) $\wedge$ (moreEquSpecific(TU_{fl}, TU)) **then**
    if (merge(TU, TU_{fl}) $\neq$ **null**) **then**
      return $\langle$0.8 - distanceFactor(TU), merge(TU, TU_{fl})$\rangle$
    else fail
  else fail
% *Ex:* *"Let's meet Tuesday." resolved to Tue 24 Sept*
"How about 2?" resolved to 2pm Tue 24 Sept
"On Tuesday?" resolved to Tue 24 Sept
This subcase differs from the previous subcase in that in this case a relationship can be made to hold if the current utterance is less specific than the one on the focus list.

Anaphoric Rule 3: *(A type of relation* **A-frame-of-reference***)*
A forward time is calculated with respect to a time on the Focus List as a frame of reference.

Subcase i:
  if not isDeictic(TU) **then**
    if {**forward-looking-adjective**} $\cap$ *ILT*$\rightarrow$*SpanUtt* $\neq$ {} **then**
      if(leastSpecific($TU \rightarrow$ *startFields*) $\leq$ **date**) **then**
        for each non-empty Temporal Unit TU_{fl} from FL, starting with most recent
          if ($TU \rightarrow$ *name* = {**week**} **then**
            Base_CF = 0.4
          else
            Base_CF = 0.3
          if moreEquSpecific(TU, TU_{fl}) **then**
            RF = retrieveStartDate(TU_{fl})
            new_TU = next(mostSpecific(start,TU), RF)

return $\langle$*Base_CF* - distanceFactor(TU), merge(TU, *new_TU*)$\rangle$
else fail
else fail
else fail
else fail
% *Ex: "Let's meet next Monday." resolved to Mon 30 Sept*
"Tuesday is better for me." resolved to Tue 1 Oct

Subcase ii:
if not isDeictic(TU) **then**
if(leastSpecific($TU \rightarrow startFields$) $\leq$ **date**) **then**
for each non-empty Temporal Unit TU_{fl} from FL, starting with most recent
if moreEquSpecific(TU, TU_{fl}) **then**
RF = retrieveStartDate(TU_{fl})
new_TU = nextInclToday(mostSpecific(start,TU), RF)
return $\langle$0.35 - distanceFactor(TU), merge(TU, *new_TU*)$\rangle$
else fail
else fail
else fail
% *This is nearly identical to the previous case, but we're not looking for a* **forward-looking-adjective**, *and today's date is considered.*
Ex: "Let's meet next Monday." resolved to Mon 30 Sept
"Monday is good." resolved to Mon 30 Sept
Since the second sentence doesn't provide clues about the Monday (such as "next", "the following" etc.), the current Monday on the focus list is used as opposed to the Monday following Monday the 30th.

Anaphoric Rule 4: *(A type of relation* **A-modify***)*
The current time is a modification of a previous time

if not isDeictic(TU) **then**
if(mostSpecific(**both**, TU) $\geq$ **time_of_day**)
for each non-empty Temporal Unit TU_{fl} from FL, starting with most recent
if (moreSpecificLow(TU, TU_{fl}) **then**
return $\langle$0.35 - distanceFactor(TU), mergeUpper(TU_{fl}, TU)$\rangle$
else fail
else fail
else fail
% *Ex: "Let's meet Tuesday at 2:00." resolved to 2 pm, Tue 24 Sept*
"3:00 is better for me." resolved to 3 pm, Tue 24 Sept

The following anaphoric rules do not appear in the paper.

Anaphoric Rule 5: *(A type of relation* **A-frame-of-reference***)*

if ($TU \rightarrow startFields$ = {}) $\wedge$ ($TU \rightarrow endFields \neq$ {}) **then**
for each non-empty Temporal Unit TU_{fl} from FL, starting with most recent
if moreEquSpecific(TU, TU_{fl}) **then**
new_TU = mergeUpper(TU_{fl}, *TU*)
if(leastSpecific(**end**, TU) $\leq$ **date**) **then**
RF = retrieveStartDate(TU_{fl})
new_TU$\rightarrow$endFields = next(mostSpecific(*endFields*, TU), *RF*)
return $\langle$0.5 - distanceFactor(TU),*new_TU*$\rangle$
else fail
else fail
else fail
% *Ex: "How about Tuesday the 25th at 2?" resolved to 2 pm, Tue 25 Sept*
"I am busy until Friday", resolved to Tue 25-Friday 28 Sept

Anaphoric Rule 6: *(A type of relation* **A-frame-of-reference***)*

Subcase i:
if not(isDeictic(TU)) **then**
if($TU \rightarrow Spec \in$ {**that, same, all_range, less_than, more_than, long**})
$\wedge$ ($TU \rightarrow Name \in$ {**month, week, day, time**}) **then**
TU_{fl} = lastFLRange(*Name*)
if (mergeUpper(TU_{fl}, *TU*)) $\neq$ **null then**
return$\langle$0.5,mergeUpper(TU_{fl}, *TU*)$\rangle$
else if (merge(TU_{fl}, *TU*))$\neq$ **null then**
return$\langle$0.5,merge(TU_{fl}, *TU*)$\rangle$
else fail
else fail
else fail
% *Ex: "Let's meet Oct 8th to Oct 11th" resolved to Tue 8 Oct - Fri 11 Oct*
"that week sounds good" resolved to Tue 8 Oct - Fri 11 Oct

Subcase ii:
if not(isDeictic(TU)) **then**
Name = retrieveField(name, *TU*)
Spec = retrieveField(specifier, *TU*)
if(**the_rest_of** $\in$ *Spec*) $\wedge$ (**that** $\in$ *Spec*) **then**
$startTU \rightarrow startFields$ = applyRule(TU, ANA3, 'i')
$endTU \rightarrow endFields$ = next(*Name*, *startFields*)
$newTU$ = merge(*startTU*, *endTU*)
else if(**the_end_of** $\in$ *Spec*) $\wedge$ (**that** $\in$ *Spec*) **then**
RF = applyRule(TU, ANA3, 'i')
$newTU$ = the_end_of(*RF*)
return $\langle$0.5,*newTU*$\rangle$
else fail
else fail

% *Ex:* *"Let's meet Oct 8th to Oct 11th" resolved to Tue 8 Oct - Fri 11 Oct*
"The end of that week is better" resolved to Thu 10 Oct - Fri 11 Oct

Anaphoric Rule 7: *(A type of relation* **A-co-reference***)*

if(**other** $\in$ *TU*$\rightarrow$*Spec*) $\wedge$ (**indefinite** not in *TU*$\rightarrow$*Spec*) **then**
  if(*Name* $\in$ {**month, week, day, time**}) **then**
    $newTU$ = lastFLRange(TU,2)
    if(merge($newTU$, TU) $\neq$ **null**) **then**
      return ⟨0.7 - distanceFactor($newTU$),merge($newTU$,TU)⟩
  else fail
else fail
% *Ex:* *"Let's meet Tuesday." resolved to Tue 24 Sept*
"How about Thursday?" resolved to Thu 26 Sept
"No, the other day sounds better" resolved to Tue 24 Sept

distanceFactor(TU_{fl}), new_TU⟩

Anaphoric Rule 8: *(A type of relation* **A-co-reference***)*

$Name :=$ retrieveField(**name**, TU)
$Spec =$ retrieveField(**specifier**, TU)
if($Name :\in$ {**week, day, time**}) $\wedge$ ($Spec \in$ {**that, plural, both_of**}) **then**
  for each non-empty Temporal Unit TU_{1-fl} from FL, starting with most recent
    for each non-empty Temporal Unit TU_{2-fl} from FL, starting with TU_{1-fl}
    if(isRange($Name :$, TU_{1-fl})) $\wedge$ (isRange($Name :$, TU_{2-fl})) **then**
      $Speaker1 =$ retrieveField(**speaker**, SpanUtt)
      $Speaker2 =$ retrieveField(**speaker**, SpanUtt)
      if($Speaker1 = Speaker2$) **then**
        if($Name =$ **day**) $\wedge$ ($TU_{1-fl} \wedge TU_{2-fl}$ are within the same week) $\vee$
        if($Name =$ **week**) $\wedge$ ($TU_{1-fl} \wedge TU_{2-fl}$ are within the same year) $\vee$
        if($Name =$ **day**) $\wedge$ ($TU_{1-fl} \wedge TU_{2-fl}$ are in the same day) **then**
          return ⟨0.7, [merge(TU, TU_{1-fl}), merge(TU, TU_{2-fl})]⟩
        else fail
      else fail
    else fail
else fail
Ex: *"How about Tuesday and Wednesday?" resolved to Tue 24 Sept, Wed 25 Sept*
"Those days sound fine." resolved to Tue 24 Sept, Wed 25 Sept
% This rule heuristically uses the last *two* occurrences of $Name :$.

2. Normalized form of a Temporal Unit

This appendix gives the specific structure, and possible information contained within a normalized *Temporal Unit* (TU), as used in the *Artwork* project, developed at New Mexico State University in conjunction with the Computing Research Laboratory. A list of Temporal Units **LTU** is a structure containing one or more TU's.

2.1 Notation Used

The following BNF-style convention is used:

1. **atom** An **atom** is a single entry in a list. An atom cannot be expanded any further.

2. <u>Non-terminal</u> A <u>Non-terminal</u> indicates a structure that is constructed by either one or more atoms, or of other non-terminals.

3. **[...]** open-close angeled brackets denote the beginning and end of a list. For example, [**this, is, a, list**] is a list containing four **atom**s.

4. * An asterisk denotes a list that may be repeated zero or more times. For example: [Repeat, this, list]* → [ ] (empty list) or [**repeat, this, list**],..., [**repeat, this, list**].

5. + A plus sign denotes a list that occurs one or more times. time.

6. | The "|" sign indicates a choice. For example: <u>Person</u> = **man** | **woman**

2.2 High Level Format of a When Frame

All structures in the Artwork project, adhere with the following format:

[<u>Structure Name</u>, <u>Structure Filler</u>]

For example,

[**Structure Name**, <u>Structure Filler</u>]
[**sday_week**, <u>SDAY_WEEK</u>]
[**sday_week**, **thursday**].

The general format of an <u>LTU</u> follows:

<u>LTU</u> = [**when-frame**, <u>TU</u>*]

Thus, the name of the structure is *when-frame* and the filler is either zero ([ ]), or more TU.

2.3 Detailed Format of a TU

```
[when,
    [connective, CONNECTIVE],
    [gen_spec, GEN_SPEC],
    [duration, [length, DURATION], [dur_specifier, DUR_SPECIFIER],
    [name, NAME],
    [interval,
        [specifier, [SPECIFIER]],
        [start,
            [sday_status, [DAY_VALUE, DAY_ORIGIN],
                [sday_week, DAY_WEEK],
                [sday, DAY],
                [stime_day, TIME_DAY],
                [sam_pm, AM_PM]],
            [smonth, [MONTH_VALUE, MONTH_ORIGIN]],
            [shour_status, [HOUR_VALUE, HOUR_ORIGIN],
                [stime_adv, TIME_ADV],
                [shour, HOUR],
                [smin, MIN]]],
        [end,
            [eday_status, [DAY_VALUE, DAY_ORIGIN],
                [eday_week, DAY_WEEK],
                [eday, DAY],
                [etime_day, TIME_DAY],
                [eam_pm, AM_PM]],
            [emonth, [MONTH_VALUE, MONTH_ORIGIN]],
            [ehour_status, [HOUR_VALUE, HOUR_ORIGIN],
                [etime_adv, TIME_ADV],
                [ehour, HOUR],
                [emin, MIN]]]],
    [modifiers, MODIFIER]]
```

Figure 1: Structural layout of a Temporal Unit.

Figure 1 illustrates the structural layout of a single TU. The following tables and figures presents the possible values each temporal field can take (Table 1, Figure 2 and Table 2). The NAME field indicates the name referred to in the fields in the above structured TU. The VALUE column illustrates the set of values that each entry might take. A **null** value indicates that no information is available. The third column provides a description of the *name* field. An in-depth description about each field is given at the end of the table. The last field in the table, "**R**" indicates whether the information can be repeated. A "**Y**" indicates that it can be, and an "**N**" indicates that it cannot be. If a field is repeated, it is preceded by the keyword “***multiple***”, otherwise it is not preceded by the keyword. For example,

dur_specifier = {***multiple*, the_end_of, second**} and
dur_specifier = {**the_end_of**}

Since dur_specifier can be repeated, it is indicated by the keyword ***multiple***, as shown in the first example. However, since there is only one filler in the second example, the keyword ***multiple*** is omitted. Fields that can take on a large number of values or need special attention are underlined in the second column of the table, and expanded at the end of this section.

A Temporal Unit comprises three parts: information about the time in general (e.g., duration, name, etc.), the start time, and the end time. Table 1 shows information on the general fields, and Figure 2 indicates the values these fields can take. Table 2 shows fields applicable to the starting time of a TU. The ending time fields are similar to the fields in Table 2, so no table is shown for the ending times.

NAME	POSSIBLE VALUES	DESCRIPTION
CONNECTIVE	**and** | **because** | **but** | **for_example** | **if** | **or** | **so** | **then** | **therefore** | **unless** | **that_is_to_say** | **null**	how multiple TUs are connected
GEN_SPEC	**generic** | **specific** | **null**	the genericity of a time
DURATION	0..MAXINT (only whole numbers) | **epsilon** | **undetermined** | **null**	the difference (in hours) between the start and end time
DUR_SPECIFIER	**determined** | **part_determined** | **not-complete**	the frame from which the duration information was retrieved unless calculated or null; repeatable
NAME	**Indexical** | **Special Name** | **Unit** | **null**	special information about the current TU not necessarily determinable by other fields.
SPECIFIER	**Specifier** | **0..MAXINT (only whole numbers)** | **null**	Info

Table 1: General fields of a Temporal Unit

Specifier ::=

all_member | **all_range** | **also** | **another** | **any** | **anytime** | **approximate** |
at_least | **behind** | **both_of** | **concrete** | **couple** | **definite** | **early** |
either_of | **even** | **exact** | **except** | **few** | **first** | **following** | **fourth** |
in_front_of | **indefinite** | **last** | **late** | **less_than** | **long** | **middle** | **more_than** |
most_member | **most_range** | **necessary** | **negative** | **next** | **only** | **other** | **particular** |
perhaps | **plural** | **preceding** | **same** | **second** | **second_last** | **short** | **some** | **sometime** |
that | **the_end_of** | **the_rest_of** | **third** | **this** | **what** | **which**

Indexical ::=

now | **today** | **tomorrow** | **date** | **time** | **then** | **that** | **there** |
here | **later** | **when** | **what**

Special Name ::=

independence_day | **thanksgiving** | **thanksgiving_day** | **labor_day**

Unit ::=

minute | **hour** | **day** | **week** | **month** | **year** |
decade | **century** | **lunch_time** | **working_day** | **weekend** | **weekday** |
early | **late** | **sometime** | **anytime**

Figure 2: Possible values for the general fields

NAME	POSSIBLE VALUES	DESCRIPTION	R
SDAY_VALUE	**determined | part_determined | undetermined | same | null**	amount of information available to determine the start date	**N**
SDAY_ORIGIN	**when | clarified | topic | null**	the frame from which the start day information was retrieved	**Y**
SDAY_WEEK	**monday,. . . , sunday | null**	the start week-day	**N**
SDAY	**1,. . . , 31 | null**	the start date	**N**
STIME_DAY	**afternoon | evening | mid_afternoon | mid_morning | midnight | morning | night | noon | null**	the general start time of a 24-hour period	**N**
SAM_PM	**am | pm | null**	the start time (morning | afternoon)	**N**
SMONTH_VALUE	**1,. . . , 12 | null**	the starting month in a numerical format. E.g. | January = 1,. . . , December = 12	**N**
SMONTH_ORIGIN	**when | clarified | topic | null**	the frame from which the start month was retrieved	**Y**
SHOUR_VALUE	**determined | part_determined | undetermined | same | null**	amount of information available to determine the start time	**N**
SHOUR_ORIGIN	**when | clarified | topic | null**	the frame from which the start hour information was retrieved	**Y**
STIME_ADV	**after | at | before | from | 'til | to | until | null**	the adverb associated with the start time	**N**
SHOUR	**1,. . . , 12 | null**	This field represents the start hour of a time.	**N**
SMIN	**0,. . . , 59**	the start minutes associated with the start hour	**N**

Table 2: Temporals fields of a Temporal Unit

2.4 Examples

This section provides examples, and a brief discussion about selected field representations given in Table 1 and Table reftemporal-fields. See Figure 3.

CONNECTIVE:

Ex: "I can meet Wednesday or Thursday"
This sentence will produce two TUs, both of which have *CONNECTIVE* *set to* **or**

GEN_SPEC:

Ex: "How about Wednesday?"
Since "Wednesday" is any wednesday, *GEN_SPEC* *is set to* **generic**

Ex: "OK, that Wednesday sounds good"
In this example, the speaker is referring to a specific Wednesday, and, thus, *GEN_SPEC* *is set to* **specific**

DURATION:

Ex: "I can meet from 2:00 until 4:00"
DURATION *is set to* **2** *(hours)*

NAME:

Ex: "I can meet next Thursday"
NAME *is set to* **next**
(It gives specific information about the time)

SPECIFIER:

Ex: "The second week in August is good for me"
In this example, *SPECIFIER* *is set to* **week**
(NAME = **second***)*

SDAY_VALUE:

Ex: "I can meet Thursday"
SDAY_VALUE *is set to* **part-determined**, *since the exact Thursday cannot be determined.*

Ex: "I can meet Thursday the 11^{th} of August"
SDAY_VALUE *is set to* **determined**, *since we have enough information about the Thursday in question*

SHOUR_VALUE:

Ex: "How about 4 or so?"
SHOUR_VALUE *is set to* **part-determined**, *since we don't know the exact time. (In this example, the minute information is missing)*

Ex: "How about next Thursday"
SHOUR_VALUE *is set to* **undetermined**, *since no information about the time is available.*

Figure 3: Examples of field values for temporal units

3. Coverage of Temporal Expressions

The BNF grammar in Figure 4 describes the core set of the temporal expressions handled by our temporal reference resolution system. To simplify the grammar, some variations are not specified. Combinations of the expressions are also allowed, provided that this doesn't lead to contradictory temporal interpretations. This is based on English transcription of the Spanish phrases that are covered by the CMU semantic parser. Note that conjunction is allowed, although not specified here. Also, brackets are used for optional components, and parentheses are used for grouping items.

```
<TEMPORAL EXPRESSION> ::= [<RELATIVE>] <DATE PHRASE>
              | [<RELATIVE>] <TIME PHRASE>
              | [<RELATIVE>] <DATE PHRASE> ["at"] <TIME PHRASE>
              | [<RELATIVE>] <TIME PHRASE> ["on"] <DATE PHRASE>
              | "from" <DATE PHRASE> ("to"|"until") <DATE PHRASE>
              | "from" <TIME PHRASE> ("to"|"until") <TIME PHRASE>

<RELATIVE> ::= "at" | "around" | "after" | "before" | "on" | "in"

<DATE PHRASE> ::= <DEICTIC> | <RELATIVE DATE> | <SPECIFIC DATE> | <ABSOLUTE DATE>

<DEICTIC> ::= "today" | "tomorrow" | "yesterday" | "now"

<RELATIVE DATE> ::= <SPECIFIER> <PERIOD>

<SPECIFIC DATE> ::= <DEMONSTRATIVE> <PERIOD>

<SPECIFIER> ::= "next" | "following" | "coming" | "the rest of"
                | "the end of" | "this" | "this coming" | "all"

<DEMONSTRATIVE> ::= "that" | "those"

<PERIOD> ::= "day" | <WEEKDAY> | "week" | <MONTH> | "month" | "morning"
             | "afternoon" | "lunchtime" | "evening" | "night"

<ABSOLUTE DATE> ::= [<WEEKDAY>] ["the"] <MONTH DATE> ["of" <MONTH>]
                    | [<WEEKDAY>] <MONTH> <MONTH DATE>
                    | ["in"] <MONTH>
                    | ["on"] <WEEKDAY>

<TIME PHRASE> ::= <HOUR_MIN> | <TIME OF DAY>

<HOUR_MIN> ::= <HOUR> <MINUTE>  [<TIME OF DAY>]

<TIME OF DAY> ::= "am" | "pm" | "in the morning" | "in the afternoon"
                  | "lunch" | "night"
```

Figure 4: BNF specification for system coverage of temporal expressions