

Capítulo 3

DETERMINACIÓN DINÁMICA DE VALORES DE VERDAD DE CONDICIONES DE REGLAS DE GENERACIÓN DE TEXTOS

Víctor M. Castel

En Víctor M. Castel, Comp. (2005) *Desarrollo, implementación y utilización de modelos para el procesamiento automático de textos*. Mendoza: Editorial de la Facultad de Filosofía y Letras, UNCuyo: 23-34.
ISBN del soporte *Internet*: 987-575-019-0

Determinación dinámica de valores de verdad de condiciones de reglas de generación de textos

Víctor M. Castel

Consejo Nacional de Investigaciones Científicas y Técnicas
Universidad Nacional de Cuyo
Mendoza, Argentina
vcastel@lab.cricyt.edu.ar

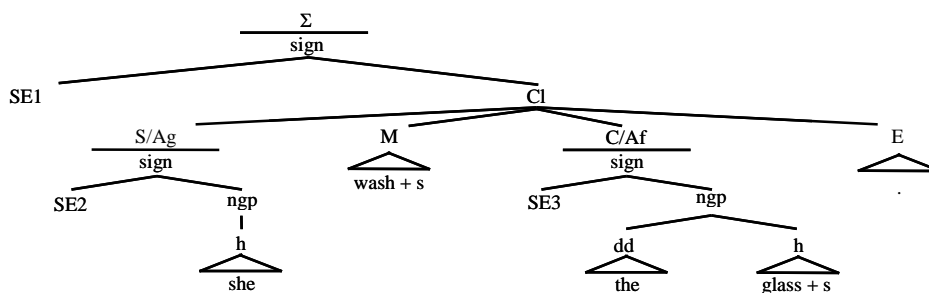
Resumen

La Léxico-Gramática de Cardiff (LGC; Fawcett 2000, Fawcett y otros 1993) es una gramática sistémica funcional orientada a la generación de textos en inglés. Las reglas semánticas y sintácticas de la LGC son implicaciones materiales, $p \rightarrow q$, en las que la condición p puede ser un rasgo, por ejemplo, *congruent situation* $\rightarrow q$, o una disyunción de rasgos, por ejemplo, *giver* \vee *seeker* $\rightarrow q$, o una conjunción de rasgos, por ejemplo, *information* \wedge \neg *future_trp* $\rightarrow q$. Los términos de las condiciones disyuntivas pueden ser también conjunciones, y los términos de las condiciones conjuntivas pueden ser también disyunciones. Algunas de las reglas de la LGC son muy complejas porque (i) contienen condiciones con términos que son conjunciones con términos que son disyunciones, y (b) tanto los términos como las condiciones pueden estar negados. Las reglas de la LGC realizan las operaciones definidas en q (inserción de rasgos semánticos, composición, llenado, exponencia, etc.), si la condición p es verdadera en relación con una representación semántica dada. Este trabajo presenta un procedimiento computacional que permite determinar el valor de verdad de p , verdadero o falso, de manera dinámica a partir de los rasgos semánticos elegidos en el proceso de generación, y del subconjunto relevante de ausencias de rasgos que resulta de esas elecciones.

1 Introducción

La LGC es una gramática orientada a la generación de textos capaz de definir representaciones lingüísticas del tipo (1) para oraciones como (2):

(1)



(2) She washes the glasses.

Abreviaturas: Σ = variable que abarca categorías de género; Cl = clause; S = subject; Ag = agent; ngp = nominal group; h = head; M = main verb; C = complement; Af = affected; dd = deictic determiner; E = ender; SE = selection expression; SE1 = [entity, situation, ..., present trp, ..., washing, ..., outsider sth, count sth, singular sth, ...]; SE2 = [entity, thing, outsider, recoverable thing, ..., singular tc, human tc, female tc, ...]; SE3 = [entity, thing, ..., outsider, ..., artefact, container, glass c, count cc, plural cc, ...], donde *situation*, *thing*, etc. son rasgos semánticos.

Las reglas semánticas y sintácticas involucradas en la generación de (1-2) son todas implicaciones que pueden representarse, leerse e interpretarse como en (3i-iii), respectivamente:

- (3i) $p \rightarrow q$
 (3ii) si p , entonces q
 (3iii) si p es verdadera, entonces ejecute q ,

donde p y q son variables que abarcan condiciones y consecuencias, respectivamente. Sirvan los ejemplos de (4) para ilustrar valores posibles de la condición p :

- (4i) $r1 \rightarrow q$
 (4ii) $r2 / r3 / r4 \rightarrow q$
 (4iii) $r5 / r6 / (r7 \& r8) \rightarrow q$
 (4iv) $r9 \& r10 \& r11 \rightarrow q$
 (4v) $r12 \& (r13 / r14) \& r15 \rightarrow q$
 (4vi) $r16 / r17 / (r18 \& \text{not } r19) \rightarrow q$
 (4vii) $r20 \& \text{not } (r21 / r22 / r23) \rightarrow q$
 (4viii) $\text{not } (r24 / r25) \rightarrow q$,

donde "r1", "r2", etc. son rasgos semánticos, y "/" y "&" son los símbolos utilizados por la LGC para representar los operadores lógicos " \vee " y " \wedge ", respectivamente. La condición p puede ser simple, como en (4i), disyuntiva, como en (4ii-iii, vi, viii) o conjuntiva, como en (4iv-v, vii). Observemos que las condiciones disyuntivas y conjuntivas pueden contener términos que son conjunciones y disyunciones, respectivamente. Así, por ejemplo, la condición disyuntiva (4iii) contiene el término "(r7 & r8)", que es una conjunción de términos. Por su parte, la condición conjuntiva (4v) contiene el término "(r13 / r14)", que es una disyunción de términos. Los ejemplos (4vi-viii) nos muestran que tanto los términos como las condiciones pueden estar negados. En la LGC, existen reglas sintácticas, pero no reglas semánticas, con términos o condiciones negados.

Razones de espacio nos impiden desplegar las ideas fundamentales sobre los valores posibles de la consecuencia q en la LGC. Baste decir que, en las reglas semánticas, la consecuencia q es la operación que permite introducir rasgos semánticos en la representación lingüística de modo de obtener representaciones semánticas del tipo ilustrado por SE1-SE3 en (1). En las reglas sintácticas, la consecuencia q puede ser una operación de llenado, composición o exposición, entre otras operaciones posibles; así, por ejemplo, en (1), las reglas sintácticas han permitido que tres ocurrencias de la unidad *sign* llenen los elementos Σ , S/Ag , y C/Adj ; que los elementos *dd* y *h* compongan la unidad *ngp*; y que los items *she*, *the*, y "." expongan los elementos *h*, *dd* y *E*, respectivamente.

Lo esencial para los fines de este trabajo es mostrar cómo se determina el valor de verdad, verdadero o falso, de la condición p . Para ello, presentamos a continuación los aspectos centrales de un algoritmo que permite hacer esa determinación en el contexto de la LGC.

2 Algoritmo para la determinación del valor de verdad de condiciones de reglas

Esta sección está dedicada a presentar de manera informal las propiedades distintivas de los procedimientos computacionales involucrados en la determinación del valor de verdad de las condiciones de las reglas semánticas y sintácticas de la LGC.

2.1 Representaciones semánticas y condiciones de reglas

El efecto conjunto de las reglas semánticas y sintácticas es una representación lingüística del tipo ilustrado en (1). La tarea específica de las reglas semánticas es construir representaciones semánticas como SE1, SE2 y SE3. La tarea específica de las reglas sintácticas es construir uni-

dades sintácticas del tipo *Cl*, *ngp*, etc. La(s) operación(es) definida(s) en la consecuencia *q* se ejecuta(n) si la condición *p* es verdadera con respecto a una representación semántica. La representación lingüística (1) contiene las abreviaturas SE1, SE2 y SE3 de las representaciones semánticas asociadas a la unidad *Cl*, la unidad *ngp* sujeto y la unidad *ngp* complemento.

Lo fundamental a los fines de este trabajo es que el valor de verdad de la condición *p* se determina sobre la base de representaciones semánticas, es decir, conjuntos de rasgos semánticos como los ejemplificados en (5):¹

- (5i) [r1, r3, r9, r10, r11, r12, r28, r31]
- (5ii) [r7, r8, r329, r74]
- (5iii) [r12, r15, r77, r128]
- (5iv) [r18, r523, r276]
- (5v) [r20, r48, r134]
- (5vi) [r20, r21, r134]

Observemos que algunos de los rasgos semánticos de las representaciones en (5) son idénticos a términos de las condiciones de las reglas en (4), y que otros no lo son. Esta distinción es crucial para determinar el valor de verdad de las condiciones de las reglas en (4). Veamos algunos casos.

Diremos, por ejemplo, que las condiciones de las reglas (4i-ii; iv) son verdaderas con respecto a (5i), y falsas con respecto a (5ii-vi). Veamos por qué. La condición de (4i) es verdadera con respecto a la representación (5i) porque consta de un término, "r1", que corresponde a un rasgo semántico presente en la representación (5i) que es idéntico a ese término. Por su parte, la condición disyuntiva de (4ii) es verdadera con respecto a la representación (5i) porque uno de sus términos, "r3", es idéntico a un rasgo semántico presente en (5i). En cuanto a la condición conjuntiva de (4iv), diremos que es verdadera con respecto a (5i) porque para cada uno de sus términos, "r9", "r10" y "r11", hay en (5i) un rasgo semántico idéntico a él. La condición de (4i) es falsa con respecto a (5ii-vi) porque en ninguna de estas representaciones semánticas hay un rasgo que sea idéntico al término "r1". La condición disyuntiva de (4ii) es falsa con respecto a (5ii-vi) porque en ninguna de estas representaciones hay un rasgo que sea idéntico a alguno de los términos de la condición disyuntiva. La condición conjuntiva de (4iv) es falsa con respecto a (5ii-vi) porque ninguna de estas representaciones contiene tres rasgos semánticos que sean idénticos a los términos de la condición conjuntiva. La condición disyuntiva de (4iii) es verdadera con respecto a la representación semántica (5ii) porque uno de sus términos, la conjunción "r7 & r8", es verdadero con respecto a (5ii); en efecto, la conjunción "r7 & r8" es verdadera con respecto a (5ii) porque para cada uno de sus términos hay un rasgo en (5ii) que es idéntico a él. La condición de (4vii) es verdadera con respecto a (5v) y falsa con respecto a todas las demás representaciones semánticas de (5). La condición conjuntiva de (4vii) es verdadera con respecto a (5v) porque sus dos términos los son. Por un lado, el término "r20" es idéntico a un rasgo semántico de (5v). Por otro lado, el término disyuntivo "not (r21 / r22 / r23)" es verdadero porque (i) la disyunción "(r21 / r22 / r23)" es falsa con respecto a (5v), pero (ii) al estar negada se "vuelve" verdadera; la intuición es que es verdad que no hay ningún rasgo semántico en (5v) que sea idéntico a alguno de los términos de la disyunción "(r21 / r22 / r23)". Dejamos como ejercicio para el lector la determinación del valor de verdad del resto de las condiciones en relación con las representaciones semánticas de (5).

¹Usamos ejemplos abstractos porque en la LGC los rasgos semánticos son largos y las reglas completas son muy complejas, lo cual atenta contra el espacio disponible. Lo importante, sin embargo, es que la discusión presentada y los procedimientos definidos valen independientemente de la extensión y complejidad de reglas reales.

2.2 Objetivo del algoritmo de determinación de valores de verdad

El objetivo del algoritmo es determinar el valor de verdad, verdadero o falso, de las condiciones de reglas semánticas y sintácticas de la LGC. Esta determinación puede hacerse de manera dinámica en el proceso de generación, a saber: a medida que se va construyendo la representación semántica. El algoritmo se basa en las elecciones de rasgos semánticos realizadas por el usuario del generador o por el generador mismo, lo que nos permite distinguir dos tipos de rasgos: presentes y ausentes. Un rasgo semántico presente es un rasgo que ha sido elegido en el proceso de generación de una representación semántica y, por tanto, es parte de esa representación semántica. Un rasgo ausente es todo rasgo que no ha sido elegido en el proceso de generación y, por tanto, no es parte de la representación semántica.

2.3 Determinación de valores de verdad por presencia de rasgos

El algoritmo utiliza la presencia de rasgos para asignar valores de verdad a las condiciones de reglas semánticas y sintácticas. Esta asignación se hace a medida que se eligen los rasgos en el proceso de generación de la representación semántica. La idea básica es determinar el impacto que tiene la elección de un rasgo sobre el valor de verdad de todas las condiciones en las que el rasgo elegido es idéntico a uno de los términos que la componen.

El algoritmo de determinación dinámica de valores de verdad a partir de rasgos elegidos se funda en la observación de que, al no existir llamado explícito de reglas semánticas, es, en principio, ineficiente evaluar las condiciones de **todas** las reglas semánticas, ya que la mayoría de ellas resultarán falsas. Pero, dado que las condiciones de las reglas semánticas no contienen términos ni condiciones negados, es posible calcular la verdad de esas condiciones composicionalmente a partir de los rasgos elegidos en el proceso de generación. Se evita así evaluar las condiciones de todas las reglas semánticas, ya que al predeterminar cuáles son verdaderas es innecesario manipular las falsas.

El rasgo elegido puede ser idéntico o no al término de una condición simple y/o una condición conjuntiva y/o una condición disyuntiva. Puesto que tanto los términos como las condiciones de regla pueden estar negados, el impacto del valor de verdad de un término sobre las condiciones de la que es parte depende de tres factores: (i) el valor de verdad "inherente" del término, (ii) el carácter negado o no del término, y (iii) el carácter negado o no de la condición de la que es parte el término. Los tres factores permiten definir una función como la siguiente, capaz de devolver el valor de verdad aportado por un término a la condición de la que es parte:

TermTV (TNeg, CNeg, InTV),

donde *TNeg* = *TermNegation*, *Cneg* = *ConditionNegation* e *InTV* = *InherentTruthValue*. Los valores posibles de las variables *Tneg* y *Cneg* son "sí" y "no". Los valores posibles de *InTV* son *Verdadero* y *Falso*. Los valores devueltos por la función son *Verdadero* o *Falso*, según resulte el cálculo. Véase en el Anexo el procedimiento que implementa la función.

Los procedimientos involucrados en la determinación de valores de verdad de condiciones a partir de la presencia de rasgos son los siguientes:

```
EvaluatePresentTermTVImpactOnConjunction strFeature, InTV
EvaluatePresentTermTVImpactOnDisjunction strFeature, InTV
EvaluatePresentTermTVImpacToSimpleCondition strFeature, InTV
```

El valor inicial de la variable *strFeature*, es decir, el valor con el que se inicia la aplicación de estos tres procedimientos, es siempre el rasgo elegido. Los valores sucesivos de la variable *strFeature* en el ciclo de aplicación basado en ese valor inicial son términos complejos, a saber:

disyunciones en el procedimiento *EvaluatePresentTermTVImpactOnConjunction* y conjunciones en el procedimiento *EvaluatePresentTermTVImpactOnDisjunction*.

Supondremos que el valor inicial de la variable *InTV*, esto es, el valor con el que se inicia la aplicación de estos tres procedimientos, es *inherentemente verdadero*, en el sentido de que es verdad que ese término que es idéntico al rasgo elegido es parte de la representación semántica en construcción. Los valores sucesivos de *InTV* en el ciclo de aplicación de los procedimientos de determinación por presencia serán *verdadero* o *falso* según corresponda de acuerdo con el valor de verdad que asigne la función *TermTV* (*TNeg*, *Cneg*, *InTV*) a los términos conjuntivos y disyuntivos. Veamos ahora cada uno de estos procedimientos por separado.

2.3.1 EvaluatePresentTermTVImpactOnConjunction (strFeature, InTV)

El procedimiento busca en la tabla correspondiente todas las condiciones conjuntivas de reglas semánticas y sintácticas en las que *strFeature* es un término. Observemos que la LGC puede contener muchas condiciones conjuntivas diferentes con términos idénticos a *strFeature*.

Los valores de *strFeature* pueden ser un término simple o un término disyuntivo. Un término simple es un rasgo semántico. Un término disyuntivo es una condición disyuntiva. Los valores de *InTV*, verdadero o falso, se utilizan para determinar el aporte en valor de verdad del término *strFeature* al valor de verdad de la condición conjuntiva de la que es parte. Este aporte se calcula mediante la función *TermTV* (*TNeg*, *CNeg*, *InTV*), que devuelve verdadero o falso, según corresponda (véase el Anexo).

El procedimiento se activa siempre a partir de la elección de un rasgo semántico por parte del usuario del generador o por el generador mismo. Así, el valor inicial de *strFeature* es un término simple y el valor inicial de *InTV* es verdadero si aceptamos la intuición básica expuesta en §2.3 de que es verdad que el rasgo elegido forma parte de la representación semántica.

En las sucesivas aplicaciones del procedimiento que sean consecuencia del ciclo de aplicación activado por el rasgo elegido, el valor de *strFeature* será siempre un término disyuntivo pasado mediante la variable correspondiente de este mismo procedimiento llamado en *DetermineCondTV* (véase en el Anexo el procedimiento implementado); por su parte, el valor de *InTV* de ese término disyuntivo será verdadero o falso según se calcule también al interior de *DetermineCondTV*.

2.3.2 EvaluatePresentTermTVImpactOnDisjunction (strFeature, InTV)

El procedimiento busca en la tabla correspondiente todas las condiciones disyuntivas de reglas semánticas y sintácticas en las que *strFeature* es un término. La LGC puede contener muchas condiciones disyuntivas diferentes con términos idénticos a *strFeature*.

Los valores de *strFeature* pueden ser un término simple o un término conjuntivo. Un término conjuntivo es una condición conjuntiva. Los valores de *InTV*, verdadero o falso, se utilizan para determinar el aporte en valor de verdad del término *strFeature* al valor de verdad de la condición disyuntiva de la que es parte. Este aporte se calcula mediante la función *TermTV* (*TNeg*, *CNeg*, *InTV*), que devuelve verdadero o falso, según corresponda (véase el Anexo).

El procedimiento se activa siempre a partir de la elección de un rasgo semántico. Así, el valor inicial de *strFeature* es un término simple y el valor inicial de *InTV* es verdadero si aceptamos la idea básica expuesta en 2.3 de que es verdad que el rasgo elegido forma parte de la representación semántica.

En las sucesivas aplicaciones del procedimiento que sean consecuencia del ciclo de aplicación activado por el rasgo elegido, el valor de *strFeature* será siempre un término conjuntivo pasado mediante la variable correspondiente de este mismo procedimiento llamado en *DetermineCondTV* (véase el Anexo); por su parte, el valor de *InTV* de ese término conjuntivo será verdadero o falso según se calcule también al interior de *DetermineCondTV*.

2.3.3 EvaluatePresentTermTVImpactoSimpleCondition strFeature, InTV

Este procedimiento es muy sencillo y podemos, dadas las limitaciones de espacio, ignorarlo.

2.4 Determinación de valores de verdad por ausencia de rasgos

El algoritmo utiliza la ausencia de rasgos para asignar valores de verdad a las condiciones de reglas sintácticas explícitamente llamadas para aplicación por el generador. La restricción de la determinación por ausencia a sólo reglas sintácticas llamadas explícitamente obedece a la necesidad de no evaluar innecesariamente el gran número de reglas sintácticas de la LGC que no son relevantes en relación con una representación semántica determinada. Es claramente más eficiente un procedimiento de determinación que se aplica solamente a reglas llamadas explícitamente que uno que se aplica a todas las reglas sintácticas de la LGC.

La determinación de valores de verdad por ausencia es necesaria para las condiciones de reglas sintácticas, pero no para las condiciones de reglas semánticas. Si bien tanto las operaciones de las reglas sintácticas como las operaciones de las reglas semánticas se ejecutan cuando las condiciones asociadas son verdaderas, hay dos razones esenciales para esta distinción en el alcance de la determinación por ausencia.

En primer lugar, las condiciones de reglas sintácticas, pero no las condiciones de reglas semánticas, pueden estar negadas y/o pueden contener términos negados. Un término ausente negado puede hacer que una condición de regla sintáctica sea verdadera y por tanto aplicable en el proceso de generación. Al carecer de términos y condiciones negados, las condiciones de reglas semánticas nunca pueden resultar verdaderas sobre la base de la ausencia de rasgos en la representación semántica en construcción. Toda condición de regla semántica que contenga términos idénticos a rasgos ausentes de la representación semántica es falsa. Este carácter de condición falsa se sigue, digamos, por omisión: las condiciones de reglas semánticas no manipuladas por la determinación de valores de verdad por presencia son todas falsas; nuestro procedimiento de determinación por presencia simplemente no hace nada con ellas, y no es necesario aplicarles el procedimiento de determinación por ausencia.

En segundo lugar, las reglas sintácticas, pero no las reglas semánticas, pueden ser de la forma siguiente:

(6)

$cond1 \rightarrow oper1, Else\ oper2$

En una regla sintáctica como (6), si la condición *cond1* es verdadera en relación con una representación semántica determinada, entonces debe llevarse a cabo la operación *oper1*; pero si la condición *cond1* es falsa en relación con una representación semántica determinada, entonces es verdadera la condición *Else* y, por tanto, debe llevarse a cabo la operación *oper2*. Luego, está claro que es imprescindible definir explícitamente también las condiciones falsas de las reglas sintácticas para que la condición *Else* funcione apropiadamente.

Un término de una condición simple, disyuntiva o conjuntiva de una regla sintáctica llamada explícitamente puede ser idéntico a un rasgo ausente de la representación semántica construida, esto es, a un rasgo que no ha sido evaluado por los procedimientos de determinación por presencia. El objetivo del algoritmo de determinación de valores de verdad por ausencia es precisamente determinar el impacto que tiene un término ausente sobre el valor de verdad de todas las condiciones de la que es parte. Puesto que tanto los términos ausentes como las condiciones de las que son parte pueden estar negados, el impacto del valor de verdad de un término ausente sobre las condiciones de la que es parte depende también de los tres factores señalados arriba en relación con la determinación por presencia: (i) el valor de verdad "inherente" del término ausente, (ii) el carácter negado o no del término ausente, y (iii) el carácter negado o no de la condición de la que es parte el término ausente. De modo entonces que podemos utilizar la misma

función *TermTV* (*TNeg*, *CNeg*, *InTV*) definida arriba (véase el Anexo). La diferencia radica en que ahora la idea básica es que el término ausente es *inherentemente falso* y, por tanto, el valor inicial de *InTV* es también *Falso*, en el sentido de que es falso que haya un rasgo en la representación semántica construida que sea idéntico al término.

Los procedimientos utilizados para la determinación de valores de verdad por ausencia de rasgos en la representación semántica son los siguientes:

```
EvaluateAbsentTermTVImpactOnConjunction Cond
EvaluateAbsentTermTVImpactOnDisjunction Cond
EvaluateAbsentTermTVImpactOnSimpleCondition Cond
```

2.4.1 EvaluateAbsentTermTVImpactOnConjunction (Cond)

El valor de *Cond* es una condición conjuntiva de una regla sintáctica llamada explícitamente por el generador. La tarea del procedimiento es determinar la contribución al valor de verdad de la condición *Cond* por parte de cada uno de los términos que la componen. Un término de una condición conjuntiva puede ser simple o disyuntivo. Un término simple ausente es un rasgo semántico que no ha sido elegido en el proceso de generación de la representación semántica (y por tanto no ha sido evaluado por el procedimiento de determinación de valores de verdad por presencia). Los términos simples ausentes y los términos disyuntivos son evaluados al interior del procedimiento mediante el llamado a los procedimientos *DetermineCondTV* y *EvaluateAbsentTermTVImpactOnDisjunction*, respectivamente:

```
DetermineCondTV Cond, "conjunctive", TermTV (TNeg, CNeg, False), Term, Cneg
EvaluateAbsentTermTVImpactOnDisjunction Term
```

2.4.1.1 DetermineCondTV (Cond, CondType, TruthValue, Term, CondNeg)

Este procedimiento se ocupa de determinar el valor de verdad de la condición conjuntiva *Cond* sobre la base del valor de verdad aportado por cada uno de los términos simples ausentes que la componen. El valor aportado por cada término simple ausente lo provee la función *TermTV* (*TNeg*, *CNeg*, *Falso*). Como se puede apreciar en el Anexo, de manera general, es suficiente que uno de los términos de la condición conjuntiva *Cond* aporte el valor *Falso* para que la *Cond* sea Falsa. Por otra parte, *Cond* es verdadera si todos sus términos aportan el valor *Verdadero*.

2.4.1.2 EvaluateAbsentTermTVImpactOnDisjunction Term

Term es una variable que abarca los términos disyuntivos de la condición conjuntiva *Cond*. El objetivo de este procedimiento se define abajo en §2.4.2. Digamos aquí simplemente que por tratarse de un término que es una condición disyuntiva es necesario acceder a los términos que componen dicha condición disyuntiva para determinar su aporte al valor de verdad de la misma, de manera que una vez que éste haya sido determinado pueda utilizarse a su vez para determinar su aporte al valor de verdad de la condición conjuntiva *Cond*.

2.4.2 EvaluateAbsentTermTVImpactOnDisjunction (Cond)

El valor de *Cond* es una condición disyuntiva de una regla sintáctica llamada explícitamente por el generador. La tarea del procedimiento es determinar la contribución al valor de verdad de la condición *Cond* por parte de cada uno de los términos que la componen. Un término de una condición disyuntiva puede ser simple o conjuntivo. Un término simple ausente es un rasgo semántico que no ha sido elegido en el proceso de generación de la representación semántica (y por tanto no ha sido evaluado por el procedimiento de determinación de valores de verdad por presencia). Los términos simples ausentes y los términos conjuntivos son evaluados al interior

del procedimiento mediante el llamado a los procedimientos *DetermineCondTV* y *EvaluateAbsentTermTVImpactOnConjunction*, respectivamente:

DetermineCondTV Cond, "disjunctive", TermTV (TNeg, CNeg, False), Term, Cneg
EvaluateAbsentTermTVImpactOnConjunction Term

2.4.2.1 DetermineCondTV(Cond, CondType, TruthValue, Term, CondNeg)

Este procedimiento se ocupa de determinar el valor de verdad de la condición disyuntiva *Cond* sobre la base del valor de verdad aportado por cada uno de los términos simples ausentes que la componen. El valor aportado por cada término simple ausente lo provee la función *TermTV* (*TNeg*, *CNeg*, *False*). De acuerdo con la implementación (véase el Anexo), si el valor de *TruthValue* es *Verdadero* y la condición disyuntiva no está negada ("a/b/c"), la condición *Cond* resulta verdadera; la determinación concluye, ya que incluso si los demás términos fueran falsos, la *Cond* seguiría siendo verdadera. Una vez que una condición disyuntiva no negada ha recibido una asignación verdadera, la condición permanece como tal, esto es, un término falso no puede volverla falsa.

Si el valor de *TruthValue* es verdadero y la condición disyuntiva *Cond* está negada ("not (a/b/c)"), debe agregarse 1 al campo que almacena la cantidad de valores verdaderos de la condición disyuntiva *Cond*. La condición disyuntiva *Cond* es verdadera si todos los términos disyuntivos que la componen aportan el valor verdadero.

Si el valor de *TruthValue* es falso y la condición disyuntiva *Cond* no está negada ("a/b/c"), debe asignarse el valor 0 al campo que almacena la cantidad de valores verdaderos de la condición disyuntiva *Cond*. Esto garantiza que el número de términos de *Cond* nunca será idéntico al número de elementos verdaderos y en consecuencia la condición *Cond* resulta falsa o bien es evaluada nuevamente por el código precedente, en cuyo caso *Cond* resulta verdadera.

Si el valor de *TruthValue* es falso y la condición disyuntiva *Cond* está negada ("not (a/b/c)"), *Cond* resulta falsa. La evaluación puede concluir aquí, ya que incluso si los demás términos de la disyunción *Cond* son verdaderos, la condición *Cond* seguirá siendo falsa; una vez que una disyunción negada ha sido evaluado como falsa, permanece como tal, esto es, un término verdadero no puede volverla verdadera. Dicho de otra manera, es suficiente que un término pase el valor falso a una condición disyuntiva para que la condición sea falsa.

2.4.2.2 EvaluateAbsentTermTVImpactOnConjunction Term

Term es una variable que abarca los términos conjuntivos de la condición disyuntiva *Cond*. El objetivo de este procedimiento se definió arriba en §2.4.1. Agreguemos aquí que por tratarse de un término que es una condición conjuntiva es necesario acceder a los términos que componen dicha condición conjuntiva para determinar su aporte al valor de verdad de la misma, de manera que una vez que éste haya sido determinado pueda utilizarse a su vez para determinar su aporte al valor de verdad de la condición disyuntiva *Cond*.

2.4.3 EvaluateAbsentTermTVImpactOnSimpleCondition (Cond)

Por razones de espacio ignoramos este procedimiento que, de todos modos, es muy sencillo.

3 Conclusión y perspectivas

El trabajo ha presentado los aspectos esenciales de un algoritmo de determinación de valores de verdad de condiciones de reglas semánticas y sintácticas de la LGC. La propiedad distintiva del algoritmo es la manera dinámica en que se lleva a cabo esa determinación: a medida que el usuario del generador o el generador mismo elige un rasgo semántico en el proceso de construcción de la representación semántica, se evalúa toda condición de regla semántica y sintáctica que tenga un término idéntico al rasgo elegido. Los rasgos ausentes, esto es, los no elegidos, son

también relevantes para la determinación del valor de verdad de las reglas sintácticas. El objetivo es establecer el aporte en valor de verdad del término, presente o ausente, al valor de verdad de la condición de la que es parte, de manera que finalmente, en el caso de condiciones disyuntivas y conjuntivas, se pueda determinar composicionalmente el valor de verdad de la condición. El algoritmo presentado es una simplificación conceptual de los procedimientos que hemos implementado en Visual Basic 6.0 para la versión *mini* de la LGC.

En Castel y Diblasi (en preparación) se compara la eficiencia de este algoritmo con otro cuya determinación de los valores de verdad se hace de manera no-dinámica, a saber: (i) para cada regla semántica de condición disyuntiva o conjuntiva se debe determinar el carácter verdadero o falso de la misma en relación con la representación semántica en construcción (para lo cual es necesario evaluar todos los términos que componen la condición), y (ii) para cada regla sintáctica llamada se debe determinar el valor de verdad de su(s) condición(es) en relación con la representación semántica construida (para lo cual también es necesario evaluar todos los términos que componen las condiciones).

Referencias

- Víctor M. Castel y Ángela Diblasi. En preparación. Algoritmos alternativos para la determinación de valores de verdad de condiciones de reglas en la Léxico-Gramática de Cardiff: ¿Del rasgo a la condición o de la condición al rasgo? InCiHuSA, CONICET, Mendoza.
- Robin P. Fawcett. 2000. *A theory of syntax for systemic functional linguistics*. John Benjamins, Amsterdam.
- Robin P. Fawcett, Gordon H. Tucker, y Yuen Q. Lin. 1993. How a systemic functional grammar works: The role of realization in realization. In Horacek y Zock (1993: 114-86).
- Helmut Horacek y Michael Zock. Eds. 1993. *New concepts in natural language generation*. Pinter, London.

Anexo

```

Function TermTV (TNeg, CNeg, InTV)
  Select Case InTV
    Case Is = False
      If TNeg = "no" Then
        If CNeg = "no" Then
          TermTV = False
        Else
          TermTV = True
        End If
      Else
        If CNeg = "no" Then
          TermTV = True
        Else
          TermTV = False
        End If
      End If
    Case Is = True
      If TNeg = "no" Then
        If CNeg = "no" Then
          TermTV = True
        Else
          TermTV = False
        End If
      Else
        If CNeg = "no" Then
          TermTV = False
        Else
          TermTV = True
        End If
      End If
  End Select
End Function
Sub DetermineCondTV(strMatrixCondition, strTypeOfCondition, infTrueElement, strChosenFeature, strConditionNegation)
  With rsOstensiveListOfConditions
    If RecordCount > 0 Then
      .MoveFirst
      .Find ("Condition = " & strMatrixCondition & "")
      If .EOF() = False Then
        If Trim(.Fields.Item("Already_Evaluated").Value) = "yes" Then
          Exit Sub
        End If
      End If
    End With
End Sub

```

```

Else
  Select Case strTypeOfCondition
  Case Is = "simple"
    .Fields.Item("Number_Of_True_Elements").Value = intTrueElement
    If intTrueElement = 1 Then
      MarkComplexCondSemanticRuleForApplication strMatrixCondition
    End If
  Case Else
    Select Case strTypeOfCondition
    Case Is = "disjunctive"
      Select Case intTrueElement
      Case Is = True
        If strConditionNegation = "no" Then
          .Fields.Item("Already_Evaluated").Value = "yes"
          .Fields.Item("Number_Of_True_Elements").Value = 1
          MarkComplexCondSemanticRuleForApplication strMatrixCondition
          EvaluatePresentTermTVImpactOnConjunction strDisjunctiveTerm, "true"
        Else
          .Fields.Item("Number_Of_True_Elements").Value = _
            .Fields.Item("Number_Of_True_Elements").Value + 1
          .Fields.Item("Number_Of_Evaluated_Elements").Value = _
            .Fields.Item("Number_Of_Evaluated_Elements").Value + 1
          If Trim(.Fields.Item("Number_Of_Elements").Value) = _
            Trim(.Fields.Item("Number_Of_Evaluated_Elements").Value) Then
            If Trim(.Fields.Item("Number_Of_Elements").Value) = _
              Trim(.Fields.Item("Number_Of_True_Elements").Value) Then
              .Fields.Item("Already_Evaluated").Value = "yes"
              MarkComplexCondSemanticRuleForApplication strMatrixCondition
              EvaluatePresentTermTVImpactOnConjunction strDisjunctiveTerm, "true"
            End If
          End If
        End If
      Case Is = False
        If strConditionNegation = "no" Then
          .Fields.Item("Number_Of_True_Elements").Value = _
            .Fields.Item("Number_Of_True_Elements").Value + 0
          .Fields.Item("Number_Of_Evaluated_Elements").Value = _
            .Fields.Item("Number_Of_Evaluated_Elements").Value + 1
          If Trim(.Fields.Item("Number_Of_Elements").Value) = _
            Trim(.Fields.Item("Number_Of_Evaluated_Elements").Value) Then
            If Trim(.Fields.Item("Number_Of_Elements").Value) <> _
              Trim(.Fields.Item("Number_Of_True_Elements").Value) Then
              .Fields.Item("Already_Evaluated").Value = "yes"
              EvaluatePresentTermTVImpactOnConjunction strDisjunctiveTerm, "false"
            End If
          End If
        Else
          .Fields.Item("Already_Evaluated").Value = "yes"
          .Fields.Item("Number_Of_True_Elements").Value = 0
          EvaluatePresentTermTVImpactOnConjunction strDisjunctiveTerm, "false"
        End If
      End Select
    End Select
  Case Is = "conjunctive"
    Select Case intTrueElement
    Case Is = True
      .Fields.Item("Number_Of_True_Elements").Value =
        .Fields.Item("Number_Of_True_Elements").Value + 1
      .Fields.Item("Number_Of_Evaluated_Elements").Value = _
        .Fields.Item("Number_Of_Evaluated_Elements").Value + 1
      If Trim(.Fields.Item("Number_Of_Elements").Value) = _
        Trim(.Fields.Item("Number_Of_Evaluated_Elements").Value) Then
      If Trim(.Fields.Item("Number_Of_Elements").Value) = _
        Trim(.Fields.Item("Number_Of_True_Elements").Value) Then
        .Fields.Item("Already_Evaluated").Value = "yes"
        MarkComplexCondSemanticRuleForApplication strMatrixCondition
        EvaluatePresentTermTVImpactOnDisjunction strConjunctiveTerm, "true"
      Else
        .Fields.Item("Already_Evaluated").Value = "yes"
        EvaluatePresentTermTVImpactOnDisjunction strConjunctiveTerm, "false"
      End If
    End If
    Case Is = False
      .Fields.Item("Already_Evaluated").Value = "yes"
      .Fields.Item("Number_Of_True_Elements").Value = 0
      EvaluatePresentTermTVImpactOnDisjunction strConjunctiveTerm, "false"
    End Select
  End Select
End Select
End If
End If
End With
End Sub

```