

## メッセージシステムのインクリメンタルな評価

3D-6

海尻 賢二  
信州 大学

## 1. まえがき

対話的なプログラミング環境の生成器の入力記述言語として属性文法は重要なものであるが、種々の欠点をもっている。これを補うものとしてメッセージシステムが提案されているが、筆者らは実用的な入力記述言語としての改良、その効率的なインクリメンタル評価アルゴリズムの提案、そしてメッセージシステムに基づく構文エディタ生成器の実現を行っている<sup>[1]</sup>。本稿ではメッセージシステムとそのインクリメンタルな評価アルゴリズムについて述べる。

## 2. メッセージシステム

メッセージシステムにおいては属性はメッセージクラスとして表され、属性の伝播はメッセージの受渡しとして実現される。そのため属性文法のように途中の伝播だけのための規則(いわゆるコピー規則)を多く書く必要もなく、また途中でその処理は必要ない。

メッセージシステムは次の4つ組により定義する。

<G, M, S, P>

G: 核となる文脈自由文法

M: メッセージクラス

S: 意味関数

P: メッセージクラスに対する伝播順序

意味関数は次の3つがある。

[生成関数]

GENERATE <target-class> WITH <with-class-list> <body>

[受理関数]

ACCEPT <target-class> FROM <node> AS <as-class> WITH <with-class-list> <body>

[返送関数]

REPLY <target-class> FOR <for-class> WITH <with-class-list> <body>

<body>は言語Cに基づいた記述言語により記述する。

[定理] 属性文法とメッセージシステムの記述能力は等しい。即ち属性文法での記述とメッセージシステムでの記述は相互に変換できる。

(証明の概略) 属性文法での意味関数は属性の処理と伝播の両方を記述している。そこでメッセージシステムでの各種伝播はコピー規則の組合せにより実現できる。また意味規則に基づく属性の伝播は各ノードすべてにメッセージの意味関数を持たすことによりシミュレートできる。[終]

## 3. 依存関係

メッセージシステムの意味関数は生成規則毎に定義する。そこで生成規則毎にラベルを定義し(解析木のラベルとなる)、意味関数とラベルの対の間に依存関係を定義し、その依存関係に基づき意味関数の評価順序を決定する。まず文法上の7つの依存関係を次の様に定義する。

[定義] 受理関係1: 関数とラベルの対  $(a_1, l_1)$  と  $(a_2, l_2)$  は次の条件を満たす時、受理関係にあると言い、 $(a_1, l_1) LA_1(a_2, l_2)$  と書く。(1)  $a_1$  と  $a_2$  はラベル  $l_1$  上の受理関数である。(2)  $a_2$  の target-class または accept-class は  $a_1$  の with-class-list の要素である。[終]

[定義] 生成関係: 関数とラベルの対  $(g, l_1)$  と  $(a, l_2)$  は次の条件を満たす時、生成関係にあると言い、 $(g, l_1) LG(a, l_2)$  と書く。(1)  $a$  と  $g$  はラベル  $l_1$  上の関数である。(2)  $a$  の target-class または accept-class は  $g$  の with-class-list の要素である。[終]

[定義] 返送関係1: 関数とラベルの対  $(r, l_1)$  と  $(a, l_2)$  は次の条件を満たす時、返送関係1にあると言い、 $(r, l_1) LR_1(a, l_2)$  と書く。(1)  $r$  と  $a$  はラベル  $l_1$  上の関数である。(2)  $a$  の target-class または accept-class が  $r$  の for-class と等しい。[終]

[定義] 返送関係2: 関数とラベルの対  $(r, l_1)$  と  $(a, l_2)$  は次の条件を満たす時、返送関係2にあると言い、 $(r, l_1) LR_2(a, l_2)$  と書く。(1)  $r$  と  $a$  はラベル  $l_1$  上の関数である。(2)  $a$  の target-class または accept-class が  $r$  の with-class-list の要素である。[終]

[定義] 伝播関係: 関数とラベルの対  $(a, l_2)$  と  $(g, l_1)$  は次の条件を満たす時、伝播関係にあると言い、 $(a, l_2) LP(g, l_1)$  と書く。(1)  $a$  はラベル  $l_2$  上の、 $g$  は  $l_1$  上の関数である。(2)  $g$  の target-class と  $a$  の target-class は等しい。(3)  $l_1$  から  $l_2$  への途中に受理関数を持たない  $g$  の target-class の伝播の道がありうる。(4)  $a$  が <from node> 節を持てば、 $l_1$  から  $l_2$  への伝播の道は  $l_2$  へは <from node> から入る。[終]

[定義] 受理関係2: 関数とラベルの対  $(a_2, l_2)$  と  $(a_1, l_1)$  は次の条件を満たす時、受理関係2にあると言い、 $(a_1, l_1) A_2(a_2, l_2)$  と書く。(1)  $a_1$  はラベル  $l_1$  上の、 $a_2$  は  $l_2$  上の関数である。(2)  $a_1$  と  $a_2$  の target-class は等しい。(3)  $a_1$  は条件付き受理である。(4)  $l_1$  から  $l_2$  へ途中に受理関数を持たない  $a_1$  の target-class の伝播の道がありうる。[終]

[定義] 返送関係3: 関数とラベルの対  $(a, l_1)$  と  $(r, l_2)$  は次の条件を満たす時、返送関係3にあると言い、 $(a, l_1) LR_3(r, l_2)$  と書く。(1)  $a$  はラベル  $l_1$  上の、 $r$  は  $l_2$  上の関数である。(2)  $a$  の target-class と  $r$  の target-class は等しい。(3)  $(r, l_2) LR_1 LA_2 LP(g, l_1)$  なる生成関数  $g$  が  $l_1$  にある。[終]

これら7つの関係を総称してRGと呼ぶ。プログラムの評価にあたってはRGとプログラムより、プログラム上での依存関係を表す同様な7つの依存関係RPを計算し、それを利用する。

[定義] RP

RGの個々の要素のラベルをプログラム上でそのラベルを持つノードで置換したもので、伝播関係を満たすものをプログラム上の(関数, ノード)対間の依存関係RPと呼ぶ。[終]

RG-A<sub>2</sub>にサイクルがないならばRPにもサイクルはない。これは必要十分条件ではない。但しメッセージシステムの意味関数の正当性の目安となる。

#### 4. メッセージ伝播の評価

属性文法においては個々のノードの属性の属性値は実質的には集合であるとしても、取扱は単一の値である。メッセージシステムにおいては生成され、伝播するメッセージはメッセージクラスのインスタンスであり、例えばブロックを表すノードには個々の宣言を表すノードより、数多くの宣言を表すメッセージクラスのメッセージが到達する。意味関数はこれらの集合に対して作用することになる。そこで該当するノードまで伝播し、意味関数による処理を行っている及び意味関数によって受理されたメッセージ集合をそれぞれ保留メッセージ及び受理メッセージと呼ぶ。メッセージ伝播には保留メッセージしか必要とならないが、インクリメンタルな伝播には両方が必要となる。

##### [アルゴリズム1] メッセージ伝播

###### (1) メッセージシステムの前処理

文法に基づく依存関係RGを計算し、 $RG - A_2$ にサイクルがあるかどうか判定する。

###### (2) プログラムを解析し、解析木を作る。

(3) RGと解析木より、プログラムに基づく依存関係RPを計算する。

(4) RPを依存木として再構成し、順次依存木よりindgreeが0である要素(プログラム木のノード、意味関数)対を取り出し評価する。[終]

##### [定義] メッセージ評価の正当性

各ノードの保留メッセージ、受理メッセージ、意味関数の間に矛盾がないとき、メッセージ評価は正当であると定義する。具体的には:

(1) 各ノードの各受理関数について、保留メッセージをもとに受理関数を使って評価した結果、受理であれば、受理されたメッセージは受理メッセージ中に存在する。また受理メッセージ中に存在するメッセージについて、保留メッセージと受理メッセージに基づき、それを受理する受理関数が存在する。

(2) 各ノードの各受理関数について、保留メッセージをもとに受理関数を使って評価した結果、受理でなければ、そのメッセージの伝播順序による次のノードの保留メッセージにそのメッセージが存在する。

(3) 各ノードの生成関数について、受理メッセージをもとに評価した結果メッセージを生成すれば、そのメッセージは伝播順序による次のノードの保留メッセージ中に存在する。

(4) 各ノードの返送関数について、受理メッセージをもとに評価した結果メッセージを返送すれば、そのメッセージは返送先のノードの保留メッセージ中に存在する。

(5) 各ノードの保留メッセージは(2) - (4)のどれかに該当する。それ以外のメッセージが保留メッセージ中に存在することはない。[終]

#### 5. インクリメンタルなメッセージ伝播

以下では解析木Tの部分木T1をT2で置換したとし、 $T - T1$ をT3と呼ぶ。インクリメンタルなメッセージ伝播においては、伝播していったメッセージが到達したノードの保留メッセージと矛盾しない、また受理されたメッセージが受理メッセージと矛盾しない時その伝播はそこで終了させる。矛盾するときはさらに伝播が必要となる。ノードと関数の対に対して評価が必要かどうかを示すフラグを用意し、それが評価要である対のみ評価する。

##### [アルゴリズム2] インクリメンタルなメッセージ伝播

(1) T3+T2に対してインクリメンタルに依存木を作る。

(2) DTにおいてT2に関わる(ノード、関数)対を評価要とする。

(3) DTにおいてindgreeが0で評価要の要素をリストにつなぐ。

(4) {インクリメンタルに評価する}

```
while リストが空でない do
  リストより要素(n, f)を1つ取り出し、リストを
  更新する
```

```
case kind(f) of
  生成関数: class=f.withclass
    for すべてのACCEPT(class)の要素について do
      fを実行する
      if メッセージが生成された then
        setに生成されたメッセージを加える
      endif
    endfor
  setについて拡張伝播動作を行う
```

```
受理関数: class=f.target
  if このノードでの初めての再評価 then
    ACCEPT(class).old=ACCEPT(class)
    ACCEPT(class)=空
  endif
  for 関連するすべての保留メッセージについて do
    fを実行する
    if メッセージが受理された then
      set1に加える else set2に加える
    endif
  endfor
  if set1がACCEPT(class).oldと等しくない then
    評価フラグの更新
  endif
  ACCEPT(class)=set1
  set2について拡張伝播動作を行う
```

```
返送関数: class=f.for
  for すべてのACCEPT(class)の要素について do
    fを実行する
    if 返送メッセージが生成された then
      setに返送メッセージを加える
    endif
  endfor
  setについて拡張伝播動作を行う。
endwhile
```

拡張伝播動作においては保留メッセージに対して受理メッセージと同様の一致性の判定を行い、一致しなければ評価フラグの更新を行う。[終]

[定理] アルゴリズム3は正しくメッセージを再評価する。  
[証明略]

[定理] アルゴリズム3は再評価の必要のある(ノード、関数)対のみ再評価する。[証明略]

#### 6. まとめ

メッセージシステムの評価を依存木に基づいて行う方法及びそのインクリメンタルな評価法について述べた。メッセージシステムに基づく構文エディタ生成器のプロトタイプは完成している。それについては別の機会に発表したい。

##### 参考文献

[1] 海尻, メッセージの受渡しによる意味記述システム, 情報処理学会第36回全国大会