A Note on the FLBC Treatment of an Example X12 Purchase Order*

Steven Orla Kimbrough

February 24, 2001

*File: span-fullEDI-sok-sam.tex. Created: 1999-11-16. Thanks to Scott A. Moore for finding this example (VICS) and for many discussions.

1 The Example PO

Here, with line numbers added, is the original example purchase order.

1. `ST*850*0001`
2. `BEG*00*SA*005001234500**19980815`
3. `REF*DP*10`
4. `REF*PD*F948325`
5. `FOB*CC`
6. `CSH*Y`
7. `SAC*A*A260***15000*******02`
8. `ITD*01*3 *2**10**30`
9. `DTM*001*19981014`
10. `DTM*010*19980915`
11. `TD5*B*2*TINA`
12. `N1*BY**92*001`
13. `N1*ST**92*650`
14. `PO1**200*EA*21*QT*UP*012345000010*IN*00001`
15. `CTP*RS*RES*42`
16. `PO1**100*EA*6.66*QT*UP*012345000034*IN*00003`
17. `CTP*RS*RES*12.5`
18. `CTT*2`
19. `SE*19*0001`

2 Detailed Analysis of the PO

In the following enumerated list,

- `Major items in typewriter font indicate lines in the EDI message`
  - Sub-items in regular font indicate VICS documentation for field of the associated EDI segments (lines).
    /* Comments are placed between C/Java-style brackets. */
    Predicates in italic/math fonts indicate direct symbolization of sub-items.

1. `ST*850*0001`

   (a) 850 is the Transaction Set Identifier Code for the Purchase Order Transaction Set.

   /* The Transaction Set Identifier Code is encoded in the speech-act-verb of the FLBC; its argument is the unique ID of the utterance (see 2c below). */

   $$po(005001234500) \tag{1}$$

   (b) 0001 is the Transaction Set Control Number.

   /* The Transaction Set Control Number is control information; we ignore it here. Had we decided otherwise, then this field could be represented by the predicate expression *TransactionSetControlNumber(005001234500, 0001)*. */

2. `BEG*00*SA*005001234500**19980815`

   (a) 00 is the Transaction Set Purpose Code which indicates this is an Original purchase order.

   $$TransactionSetPurposeCode(005001234500,\ 00) \tag{2}$$

   (b) SA is the Purchase Order Type Code which indictes this is a Standalone Order.

   $$PurchaseOrderTypeCode(005001234500,\ SA) \tag{3}$$

   (c) 005001234500 is the Purchase Order Number.

   /* Already handled as an argument to *po()*, see 1a. */

   (d) 19990815 is the Purchase Order Date (August 15, 1998).

   /* We will use the FLBC predicate *Cul(,)*. */

   $$Cul(005001234500,\ 19980815) \tag{4}$$

3. `REF*DP*10`

   (a) DP is the Reference Number Qualifier which indicates the Reference Number that follows is a Department Number.

   (b) 10 is the department number assigned by the buyer.

   /* These two fields could be symbolized as a single predicate

   $$REF(005001234500,\ DP,\ 10) \tag{5}$$

   but doing so would unnecessarily bury important logico-semantic information. In fact, this segment and its fields are surprisingly complex from

a semantic perspective. The explanation for our treatment of them is best handled in a separate, more extensive, section, which now follows. We digress now in order to present results more directly later. */

3 Treatment of adjectives and other qualifiers

The underlying semantics here (segment (3)) is a form of modified modifier. We begin with a simple example and work our ways up to examples that are directly pertinent to the case at hand. Consider the simple sentence (complex):

Sentence Complex 3.1 *x is a red ball*

which can be paraphrased as "x is a ball and x is red." Here, "red" is an adjective. Under the event sementics theory developed by Parsons [**?**, **?**], sentence complex (3.1) has an underlying logico-semantic structure close to "There is a state of being red and x is a ball that is in that state," or formally:

Sentence Complex 3.2 *s is-a-state-of-being-red* $\wedge$ *x is-a-ball* $\wedge$ *In(x,s)*

Without going deeply into the merits of this analysis (see Parsons [**?**, **?**] for the original development and considerable detail, which we find persuasive), we note that (3.2) gets certain things right. Like (3.1) it implies that x is a ball and that something is red. This is as it should be; the formalization is getting at least something right.

Now consider a slightly more complex sentence:

Sentence Complex 3.3 *x is a big red ball*

which might be paraphrased as "x is ball and it is both big and red." Under our semantic theory we can analyze this as:

Sentence Complex 3.4 *s is-a-state-of-being-big* $\wedge$ *s′ is-a-state-of-being-red* $\wedge$ *In(x, s)* $\wedge$ *In(x, s′)* $\wedge$ *ball(x)*

(Those with certain philosophical tendancies might be happier with a single underlying state:

Sentence Complex 3.5 *s is-a-state-of-being-big* $\wedge$ *s is-a-state-of-being-red* $\wedge$ *In(x, s)* $\wedge$ *ball(x)*

For the present we defer discussion of such fine points.) From sentence (3.3) it follows both that x is a red ball and that x is a big ball. Our—really Parsons's—formalization captures this nicely.

Contrast now "x is a big red ball" with

Sentence Complex 3.6 *x is a light brown desk*

where "brown" qualifies "desk" and "light" qualifies "brown" rather than "desk." A light brown desk is surely brown, and probably dark rather than light (and may well be heavy). We have here the use of adjectives in what is called the *attributive* mode.[1] We could follow the suggestions of Parsons [**?**, page 194] and model "x is a light brown desk" as

Sentence Complex 3.7 *brown(s)* $\wedge$ *light(s)* $\wedge$ *In(x, s)* $\wedge$ *desk(x)*

where *brown(s)* should be read as "s is-a-state-of-being-brown," *light(s)* as "s is-a-state-of-being-light," and *In(x, s)* as "x is in state s." (s should be taken as a constant, x as a variable.)[2] But this formulation has its difficulties. Associated with x is a particular state s that is both brown and light, hence x has an indentical association with brown and with light. Parsons's move, or something like it, will work for sentences like (3.3) and will not work for sentences like (3.6). The ball is big and the ball is red, the desk is brown, but the desk is not light (in any nonmisleading sense); rather, it is light-brown.

An abstraction will set things aright, at least well enough for present purposes:[3]

Sentence Complex 3.9 *brown(s)* $\wedge$ *light(s′)* $\wedge$ *In(s, s′)* $\wedge$ *In(x, s)* $\wedge$ *desk(x)*

Paraphrased: "The desk is brown and the brown is light."

Before applying this approach to the case at hand, and to those that follow, it will be helpful to introduce another abstraction and some notational changes, as well as some generalizing discussion. "x is-a-state-of-being-red" can be abstracted or generalized to "x is a state of type y" or *StateType(x, y)* and in particular *StateType(s, red).* We might also have *StateType(s, big)* for "the state type s is big." With this move, sentence (3.4) becomes

Sentence Complex 3.10 *StateType(s, big)* $\wedge$ *StateType(s′, red)* $\wedge$ *In(x, s)* $\wedge$ *In(x, s′)* $\wedge$ *ball(x)*

More generally, the pattern is this: adjectives and other modifiers—including modifier modifiers such as "light" in "light brown" above—refer to underlying states for which the objects of the modifiers participate. That is what the theory developed here holds. We shall see how it works in practice.

Further, it will often be helpful in practice to add information about these various states. The present representation facilitates this. For example, we might add that state s is color state and state s' is a size state:

[1] See [**?**, page 194]: "Attributive constructions are those in which adjectives modify nouns." Larson and Segal [**?**, page 499] distinguish **pure intersective adjectives** from **pure attributive adjectives** but their usage differs from that of Parsons.

[2] Sentence (3.7) is actually closer to Parsons's style in [**?**]. In [**?**] he uses a formulation more like the following.

Sentence Complex 3.8 *brown(s)* $\wedge$ *light(s)* $\wedge$ *Theme(s, x)* $\wedge$ *desk(x)*

[3] The representation admittedly raises some intriguing, even vexing, metaphysical questions. For present purposes we can defer them. Let us first see how well the representation does its basic job. See, however, [**?**] for a related discussion.

Sentence Complex 3.11 *StateType(s, color)* $\wedge$ *StateType(*s'*, size)* $\wedge$ *StateType(*s^2*, big)* $\wedge$ *StateType(*s^3*, red)* $\wedge$ *In(x, s)* $\wedge$ *In(x,* s'*)* $\wedge$ *ball(x)* $\wedge$ *In(s,* s^3*)* $\wedge$ *In(*s'*,* s^2*)*

(*StateType(x, y)* should be read as "x is a state of type y.") Readers familiar with object–oriented design will see a correspondence: StateTypes correspond to attributes and their values correspond to values of attributes. Here, StateTypes are uniquely identified by their IDs, s and s' in the running examples.

Consider now a pair of tan pants with inseam length of 32 (inches). Here is our analysis:

Sentence Complex 3.12 *pants(x)* $\wedge$ *StateType(s, tan)* $\wedge$ *In(x,s)* $\wedge$ *StateType(*s^1*, inseam)* $\wedge$ *In(x,* s^1*)* $\wedge$ *StateType(*s^2*, length)* $\wedge$ *In(*s^1*,* s^2*)* $\wedge$ *Dimension(*s^2*, inch)* $\wedge$ *Unit(*s^2*, count)* $\wedge$ *Quantity(*s^2*, 32)*

Eventually, the state-abstractions have to stop and we have (we think sensibly) stopped here with predicates of common use.

One more illustration. Sometimes the measure or attribute of interest for a state is merely nominal, less well-established than physical metrics. Consider a shirt with size medium:

Sentence Complex 3.13 *shirt(x)* $\wedge$ *StateType(s, size)* $\wedge$ *In(x, s)* $\wedge$ *Value(s, medium)*

The predicate "Value" can be used as a generic for such cases.

We are now in position to apply this apparatus more or less directly to the current segment.[4]

4 Continuing

3. `REF*DP*10`

   (a) DP is the Reference Number Qualifier which indicates the Reference Number that follows is a Department Number.

   (b) 10 is the department number assigned by the buyer.

   /* Together, these yield: */

   PO Line Symbolization 4.1 $\exists s \exists s'$*(StateType(s, REF)* $\wedge$ *In(005001234500, s)* $\wedge$ *StateType(*s'*, DN)* $\wedge$ *Value(*s'*, 10)* $\wedge$ *In(s,* s'*))*

4. `REF*PD*F948325`

[4] See [**?**, page 250] for a puzzled remark on related topic: "... in explaining the use of 'smoothly' in 'the smoothly shaved leg,' or 'deftly' in 'the deftly broken egg,' ... [there is a puzzle.] If 'shaved' is an adjective in 'the smoothly shaved leg,' then some account needs to be given of the status of 'smoothly', whose normal use is to modify a verb, not a noun or adjective."

(a) DP is the Reference Number Qualifier which indicates the Reference Number that follows is a Promotion/Deal Number.

(b) F948325 is the promotion/deal number.

/* The structure here is essentially that of segment (3). */

PO Line Symbolization 4.2 $\exists s \exists s' (StateType(s, REF) \wedge In(005001234500, s) \wedge StateType(s', Promotion/DealNumber) \wedge Value(s', F948325) \wedge In(s, s'))$

/* These two fields are best symbolized as a single predicate. */

$$REF(005001234500, PD, F948325) \tag{6}$$

/* Note: Repeat use of the *REF(,,)* predicate. Note also that perhaps *ReferenceNumberQualifier(,,)* is a better name for this predicate. */

5. `FOB*CC`

(a) CC is the Shipment Method of Payment which indicates the transportation charges are Collect.

$$ShipmentMethodOfPayment(005001234500, CC) \tag{7}$$

/* Note that this applies to all items ordered. */

6. `CSH*Y`

(a) Y is the Sales Requirement Code which indicates the item(s) is to be Back Order(ed) if Out of Stock.

$$SalesRequirementCode(005001234500, Y) \tag{8}$$

/* Note that this applies to all items ordered. */

7. `SAC*A*A260***15000*******02`

(a) A is the Allowance or Charge Indicator which indicates this SAC segment identifies an Allowance.

$$StateType(s1, AllowanceChangeIndicator) \wedge \tag{9}$$
$$StateType(s2, Allowance) \wedge In(005001234500, s1) \wedge \tag{10}$$
$$In(s1, s2)$$

/* Paraphrased: "*s1* is a state of type AllowanceChangeIndicator, with *s2* a state of being an Allowance; moreover, our utterance 005001234500 is in the *s1* state which itself is in the *s2* state." Note that *s1, s2* must be unique IDs. */

(b) A260 is the Service, Promotion, Allowance, or Charge Code which indicates that this is an Advertising Allowance.

$$StateType(s3,\ ServicePromotionAllowanceOrChange) \wedge StateType(s4,\ A260) \wedge In(s2,\ s3) \wedge In(s3,\ s4) \quad (11)$$

/* *Advertising* might be clearer than *A260* but we follow the original as closely as possible. Note here that advertising (*A260*) modifies *ServicePromotionAllowanceOrChange* which modifies *Allowance,* as should be the case. */

(c) 15000 is the Amount of the advertising allowance ($150.00).

$$Dimension(s4,\ USCents) \wedge Quantity(s4,\ 15000) \wedge Unit(s4,\ count) \quad (12)$$

/* *s4* is the relevant state and now we give its measure, in terms of US cents. */

(d) 02 is the Allowance or Charge Method of Handling Code which indicates that the allowance will be reflected in the total transaction amount.

$$StateType(s5,\ AllowanceOrChangeMethodOfHandling) \wedge Value(s5,\ 02) \wedge In(s4,\ s5) \quad (13)$$

/* Note: We have dropped the existenial quantifiers over states, in favor of their skolemization in terms of constants (*s1, s2,* ...). These constant names should be unique IDs; nevertheless, for the sake of readability we shall here reuse the names segment by segement. */

8. `ITD*01*3 *2**10**30`

(a) 01 is the Terms Type Code which indicates the berms are Basic.

$$StateType(s1,\ TermsTypeCode) \wedge StateType(s2,\ Basic) \wedge In(005001234500,\ s1) \wedge In(s1,s2) \quad (14)$$

/* Paraphrased: "*s1* is a state of type TermsTypeCode, with *s2* a state of being an Basic; moreover, our utterance 005001234500 is in

the *s1* state, which itself is in state *s2*." Or, there is a state of type TermsTypeCode associated with our PO and that state has the value Basic. */

(b) 3 is the Terms Basis Date Code which indicates the Invoice Date is the beginning of the terms period.

$$StateType(s3,\ TermsBasisDateCode) \wedge Value(s3,\ 3) \wedge In(s2,\ s3) \tag{15}$$

/* Paraphrased: "The state *s2* itself is in state *s3*, whose value is 3 and type is TermsBasisDateCode." */

(c) 2 is the Terms Discount Percent available to the purchaser if an invoice is paid on or before the Terms Discount Days Due.

$$StateType(s4,\ TermsDiscountPercent) \wedge Value(s4,\ 2) \wedge In(s2,\ s4) \tag{16}$$

/* Paraphrased: "The state *s2* itself is in state *s4*, whose value is 2 and type is TermsDiscountPercent." */

(d) 10 is the Terms Discount Days Due

$$StateType(s5,\ TermsDiscountDaysDue) \wedge Value(s5,\ 10) \wedge In(s2,\ s5) \tag{17}$$

/* Paraphrased: "The state *s2* itself is in state *s5*, whose value is 10 and type is TermsDiscountDaysDue." */

(e) 30 is the Terms Net Days

$$StateType(s6,\ TermsNetDays) \wedge Value(s6,\ 30) \wedge In(s2,\ s6) \tag{18}$$

/* Paraphrased: "The state *s2* itself is in state *s6*, whose value is 30 and type is TermsNetDays." */

Note: This segment and the previous one, especially, show how even with transparent communication languages detailed knowledge of the subject matter will still be required. There is simply no way around this.

9. `DTM*001*19981014`

(a) 001 is the Date/Time Qualifier which indicates the Date that follows is the date after which the order is considered canceled (Cancel After).

(b) 19981014 is the cancel after date (October 14, 1998).

/* Placed naïvely into logic this segment has the form

CancelAfterDate(005001234500, 19981014)

Points arising:

- Much logical/semantic structure lies hidden in these representations. What about CancelOnDate and the many other obvious variants?
- The segment can't mean literally what it suggests. Not *all* purchase orders are canceled, only those that are not honored (by a certain date).

We will return to this below. For the present, three rather standard predicates are needed:

$cancel(e) \wedge DIR(e, 005001234500) \wedge Cul(e, 19981014)$

and we shall say in effect that if the purchase order is not honored before 19981014, then on 19981014, it is—we declared—canceled (by event e). DIR is the direct object of the verb "cancel".

*/

10. `DTM*010*19980915`

(a) 010 is the Date/Time Qualifier which indicates the Date that follows is the Requested Ship date.

(b) 19980915 is the requested ship date (September 15, 1998).

/* See the discussion of line (9), which applies here as well. What this segment is saying is roughly, "We request that the goods that are the subject of this po be shipped by you on or before September 15, 1998." Below, we shall refer to these goods as $g1$ and $g2$. Let $[g1, g2]$ represent a bundle consisting of these goods.

*/

$$\exists e \exists t(ship(e) \wedge SUBJ(e, \text{'Fashion Today'}) \wedge DIR(e, [g1, g2]) \wedge Cul(e, t) \wedge t \leq 19980915$$

/* Notice that nothing is said here about where the shipment goes. Such is the EDI, but we could easily add that the shipment is to the speaker (The Corner Store) by conjoing to the above expression this one:

$\wedge$ *To(e, 'The Corner Store')*

or, using the meaning of "ship"

$\wedge$ *INDIR(e, 'The Corner Store')*

(In both cases the added clauses are within the scope of the existential e.)
*/

From WordNet 1.6:

```
1 sense of ship

Sense 1
{01328437} <verb.motion> transport1#4, send#4, ship#1
   -- (transport commercially)
       => {01328337} <verb.motion> barge1#2 --
        (transport by barge on a body of water)
       => {01331285} <verb.motion> dispatch#1, despatch#1, send off#1 --
          (send off promptly)
           => {01331167} <verb.motion> bundle off#1 --
          (send off unceremoniously)
           => {01331450} <verb.motion> route2#1 --
            (send documents or materials to appropriate destinations)
       => {01331576} <verb.motion> forward#1, send on#1 --
           (send or ship onward from an intermediate post or
            station in transit; "forward my mail")
```

11. `TD5*B*2*TINA`

    (a) B is the Routing Sequence Code which indicates Origin/Delivery Carrier (Any Mode).
    StateType(s1, RoutingSequenceCode) $\wedge$ *StateType(s2, B)* $\wedge$ *In(005001234500, s1)* $\wedge$ *In(s1, s2)*

    (b) 2 is the Identification Code Qualifier which indicates that Indentification Code that follows is a Standard Carrier Alpha Code (SCAC).
    StateType(s3, IdentificationCodeQualifier) $\wedge$ *Value(s3 2)* $\wedge$ *In(s2, s3)*

    (c) TINA is the Standard Carrier Alpha Code of the carrier.
    StateType(s4, StandardCarrierAlphaCode) $\wedge$ *Value(s4, TINA)* $\wedge$ *In(s3, s4)*

12. `N1*BY**92*001`

    /* The logical structure here is much like that for line (11). */

    (a) BY is the Entity Identifier Code which indicates this N1 segment identifies the Buying Party (Purchaser).
    StateType(s1, EntityIdentifierCode) $\wedge$ *StateType(s2, B)* $\wedge$ *In(005001234500, s1)* $\wedge$ *In(s1, s2)*

(b) 92 is the Identification Code Qualifier which indicates the Identification Code that follows is Assigned by [the] Buyer or Buyer's Agent.
StateType(s3, IdentificationCodeQualifier) $\wedge$ *StateType(s4, 92)* $\wedge$ *In(s2, s3)* $\wedge$ *In(s3, s4)*

(c) 001 is the Identification Code for the buying party.
StateType(s5, BuyerIdentificationCode) $\wedge$ *Value(s5, 001)* $\wedge$ *In(s4, s5)*

13. `N1*ST**92*650`

(a) ST is the Entity Identifier Code which indicates this N1 segment identifies the Ship To Location.

(b) 92 is the Identification Code Qualifier which indicates the Indentification Code that follows is Assigned by [the] buyer or Buyer's Agent.

(c) 650 is the Identification Code for the ship to location.

This segment and the previous one (12) are quite similar in structure. We have seen one way of handling that structure. Here we consider another, a more direct way, but we shall no longer be able to avoid speech act considerations. Let e be the requested shipping event (see segment (10)). As noted above, we can consider the ship to location, x simply as the indirect object of the ship verb: *INDIR(e, x1).* What the current segment is saying is that the buyer is declaring that there is a state, s, of being the ship to location code whose value is 650, further that $x1$ is in that state.

Think of it this way. By previous arrangement, the buyer and the seller have set up tables for common use. One of them is the ship-to-table, having two columns: a location code and an actual address. In EDI messages and purchase orders in particular, the parties agree to use the codes rather than the actuall addresses. What's being said here is that the buyer is requesting that the seller ship the goods to that place indicated by the code 650 in the ship-to-table. So, in our terms we have an individual (not a state or an event) who we shall call *x1.* With *INDIR(e, x1)* the PO indicates that the shipment is to go to *x1.* Now we have to say what *x1* is. That's easy. We already have the apparatus:

StateType(s1, IdentificationCodeForTheShipToLocation) $\wedge$ *Value(s1, 650)*

14. `PO1**200*EA*21*QT*UP*012345000010*IN*00001`

/* We finally get to a segment that represents something like a line in an ordinary purchase order. This purchase order is for two distinct goods (lines) which we have previously named *g1* and *g2.* This segment and the next have to do with *g1.* The purchase order is requesting that these goods be shipped to the buyer. For the present, let the requested shipping event be named *e1* (see segment (10) in which we quantify on e). */

(a) 200 is the Quantity Ordered.
Quantity(g1 , 200)

(b) EA is the Unit or Basis for Measurement Code which indicates Each.
Unit(g1, count)
/* Note: our "count" is equivalent to "Each". */

(c) 21 is the Unit Price
UnitPrice(g1, p1) ∧ *Dimension(p1, USDollars)* ∧ *Unit(p1, count)* ∧ *Quantity(p1, 21.00)*

(d) QT is the Basis of Unit Price Code which indicates the Unit Price of $21.00 was Quoted.
/* The buyer is saying here that the seller quoted this price for the buyer. When we take up the illocutionary aspects of the purchase order in detail, we shall model this as the buyer asserting that the seller has declared (to the buyer) that this is the price. At present there is nothing to add. */

(e) UP is the Product/Service ID Qualifier which indicates the Product/Service ID that follows is a U.P.C. Consumer Package Code (1-5-5-1).
/* A now-familiar logico-semantic structure. */
StateType(s1, Product/ServiceIDQualifier) ∧ *Value(s1, UP)* ∧ *In(g1, s1)* ∧ *StateType(s2, UPCConsumerPackageCode)* ∧ *In(s1, s2)*

(f) 012345000010 is the U.P.C. number.
Value(s2, 012345000010)

(g) IN is the Product/Service ID Qualifier which indicates the Product/Service ID that follows is a Buyer's Item Number.
/* Again, the now-familiar structure. */
StateType(s3, Product/ServiceIDQualifier) ∧ *Value(s3, IN)* ∧ *In(g1, s3)* ∧ *StateType(s4, BuyersItemNumber)* ∧ *In(s3, s4)*

(h) 00001 is the buyer's catalog number.
Value(s4, 00001)

15. `CTP*RS*RES*42`

(a) RS is the Class of Trade Code which indicates Resale.
StateType(s1, ClassOfTradeCode) ∧ *Value(s1, RS)* ∧ *In(g1, s1)*

(b) RES is the Price Identifier Code which indicates the Unit Price that follows is the Resale (retail) price.
StateType(s2, PriceIdentifierCode) ∧ *In(g1, s2)* ∧ *StateType(s3, RES)* ∧ *In(s2, s3)*

(c) 42 is the retail price.
UnitPrice(s3, p2) ∧ *Dimension(p2, USDollars)* ∧ *Unit(p2, count)* ∧ *Quantity(p2, 42)*

16. `PO1**100*EA*6.66*QT*UP*012345000034*IN*00003`

    /* Logically, the same as line 14. */

17. `CTP*RS*RES*12.5`

    /* Logically, the same as line 15. */

18. `CTT*2`

    /* Control information, not covered. */

19. `SE*19*0001`

    /* Control information, not covered. */

5 The purchase order in full

It will be complex. We will show the overall structure, then do the pieces one by one. The overall structure is

$$(\alpha \wedge \Box(\beta^{+}))$$

where β^{+} means conjunctions ($\wedge$s) of one or more formulæ with structure β, i.e., with structure:

$$(\gamma \rightarrow (\delta \leftrightarrow \epsilon))$$

whose details will be presented as we go. (But see the FLBC BNF.)

We model the purchase order as a compound speech act. It is a request (that certain goods be shipped to the buyer), it is a promise (to pay for the goods, if shipped), it is an assertion of certain (puted) facts, and it is a declaration(a mak ing so by saying so) of certain facts. Roughly, there will be a β clause for each of these (sub)speech acts. We begin with the α clause. Note: no existential quantifiers, just the skolemized version with globally unique names. References to previous sections and formulæ are given in curly braces, {.} and are not part of the notation.

Note well that we have engaged in further abstraction. Event predicates, such as *ship(e)* have been abstracted to *EventType(e, ship)* and (implicit) state predicates, such as *TransactionSetPurposeCode(x,y)* have been abstracted to *StateType(s, TransactionSetPurposeCode)* $\wedge$ *Value(s, y)* $\wedge$ *In(x, s).* As a consequence of this abstraction (and the others) we have a vacabulary of basic predicates that is quite sparse indeed, yet very powerful and general. A spanning set?

5.1 α: the PO header

po(005001234500) {*1a*} $\wedge$ *Speaker(005001234500, 'The Corner Store')* $\wedge$ *Addressee(005001234500, 'Fashion Today')* $\wedge$ *Cul(005001234500, 19980815)* {*2d* }

5.2 β_1: the PO declarations

$\top \rightarrow$ *(Auth(005001234500, 1)* $\leftrightarrow$ *(*

(StateType(s11, TransactionSetPurposeCode) $\wedge$ *Value(s1, 00)* $\wedge$ *In(005001234500, s11))* $\{2b\}$ $\wedge$

(StateType(s12, PurchaseOrderTypeCode) $\wedge$ *Value(s12, SA)* $\wedge$ *In(005001234500, s12))* $\{2c\}$ $\wedge$

(StateType(s13, REF) $\wedge$ *In(005001234500, s13)* $\wedge$ *StateType(s14, DN)* $\wedge$ *Value(s14, 10)* $\wedge$ *In(s13, s14))* $\{3\}$ $\wedge$

(StateType(s15, REF) $\wedge$ *In(005001234500, s15)* $\wedge$ *StateType(s16, Promotion/DealNumber)* $\wedge$ *Value(s16, F948325)* $\wedge$ *In(s15, s16))* $\{4\}$ $\wedge$

(StateType(s17, ShipmentMethodOfPayment) $\wedge$ *In(005001234500, s17)* $\wedge$ *Value(s17, CC))* $\{5\}$ $\wedge$

(StateType(s18, SalesRequirementCode) $\wedge$ *In(005001234500, s18)* $\wedge$ *Value(s18, Y))* $\{6\}$ $\wedge$

(StateType(s19, AllowanceChangeIndicator) $\wedge$ *StateType(s110, Allowance)* $\wedge$ *In(005001234500, s19)* $\wedge$ *In(s19, s110))* $\{7a\}$ $\wedge$

(StateType(s111, ServicePromotionAllowanceOrChange) $\wedge$ *StateType(s112, A260)* $\wedge$ *In(s110, s111)* $\wedge$ *In(s111, s112))* $\wedge$

(Dimension(s112, USCents) $\wedge$ *Quantity(s112, 15000)* $\wedge$ *Unit(s112, count))* $\wedge$

(StateType(s113, AllowanceOrChangeMethodOfHandling) $\wedge$ *Value(s113, 02)* $\wedge$ *In(s112, s113))* $\wedge$

$\{8\}$*:*

((StateType(s114, TermsTypeCode) $\wedge$ *StateType(s115, Basic)* $\wedge$ *In(005001234500, s114)* $\wedge$ *In(s114,s115))* $\wedge$

(StateType(s116, TermsBasisDateCode) $\wedge$ *Value(s116, 3)* $\wedge$ *In(s115, s116))* $\wedge$

(StateType(s117, TermsDiscountPercent) $\wedge$ *Value(s117, 2)* $\wedge$ *In(s115, s117))* $\wedge$

(StateType(s118, TermsDiscountDaysDue) $\wedge$ *Value(s118, 10)* $\wedge$ *In(s115, s118))* $\wedge$

(StateType(s119, TermsNetDays) $\wedge$ *Value(s119, 30)* $\wedge$ *In(s115, s119)))* $\wedge$

$\{12\}$*:*

((StateType(s120, EntityIdentifierCode) $\wedge$ *StateType(s121, B)* $\wedge$ *In(005001234500, s120)* $\wedge$ *In(s120, s121))* $\wedge$

(StateType(s122, IdentificationCodeQualifier) $\wedge$ *StateType(s123, 92)* $\wedge$ *In(s121, s122)* $\wedge$ *In(s122, s123))* $\wedge$

(StateType(s124, BuyerIdentificationCode) ∧ Value(s124, 001) ∧ In(s123, s124))) ∧

{13}:

(StateType(s125, IdentificationCodeForTheShipToLocation) ∧ Value(s125, 650) ∧ In(005001234500, s125))

))

The second declarative associated with the po is a conditional one: If the requested shipping event, *e1*, has not happened by 19981014, then at that time the po is declared canceled. Here is the formalization

(t21 ≥ 19981014 ∧ ¬Cul(e1, t21)) → (Auth(005001234500, 2) ↔ (

EventType(e21, cancel) ∧ DIR(e21, 005001234500) ∧ Cul(e21, 19981014)))
{9}

[Note: Careful here: this should probably be expressed in terms of Hold on the resultant state of e1. Later.]

5.3 β_2: requests

⊤ → (H(005001234500, 1) ↔ (

EventType(e1, ship) ∧ SUBJ(e1, 'Fashion Today') ∧ DIR(e1, [g1, g2]) ∧ INDIR(e1, 'The Corner Store') ∧ Cul(e1, t1) ∧ t1 ≤ 19980915
{10} ∧

{11}:

(StateType(s31, RoutingSequenceCode) ∧ StateType(s32, B) ∧ In(e1, s1) ∧ In(s31, s32)) ∧

[note in the above line, the change in In from the po id to e1]

(StateType(s33, IdentificationCodeQualifier) ∧ Value(s33, 2) ∧ In(s32, s33)) ∧

(StateType(s34, StandardCarrierAlphaCode) ∧ Value(s34, TINA) ∧ In(s33, s34)) ∧

{14}:

(Unit(g1, count) ∧ UnitPrice(g1, p1) ∧ Dimension(p1, USDollars) ∧ Unit(p1, count) ∧ Quantity(p1, 21.00) ∧ StateType(s35, Product/ServiceIDQualifier) ∧ Value(s35, UP) ∧ In(g1, s35) ∧ StateType(s36, UPCConsumerPackageCode) ∧ In(s35, s36) ∧ Value(s36, 012345000010) ∧ (StateType(s37, Product/ServiceIDQualifier) ∧ Value(s37, IN) ∧ In(g1, s37) ∧ StateType(s38, BuyersItemNumber) ∧ In(s37, s38) ∧ Value(s38, 00001))) ∧

{15}:

((StateType(s39, ClassOfTradeCode) ∧ Value(s39, RS) ∧ In(g1, s39))
∧

(StateType(s310, PriceIdentifierCode) ∧ In(g1, s310) ∧ StateType(s311, RES) ∧ In(s310, s311)) ∧

(UnitPrice(s311, p2) ∧ Dimension(p2, USDollars) ∧ Unit(p2, count) ∧ Quantity(p2, 42))) ∧

{16}:

(Unit(g2, count) ∧ UnitPrice(g2, p12) ∧ Dimension(p12, USDollars) ∧ Unit(p12, count) ∧ Quantity(p12, 6.66) ∧ StateType(s45, Product/ServiceIDQualifier) ∧ Value(s45, UP) ∧ In(g2, s45) ∧ StateType(s46, UPCConsumerPackageCode) ∧ In(s45, s46) ∧ Value(s46, 012345000034) ∧ (StateType(s47, Product/ServiceIDQualifier) ∧ Value(s47, IN) ∧ In(g2, s47) ∧ StateType(s48, BuyersItemNumber) ∧ In(s47, s48) ∧ Value(s48, 00003))) ∧

{17}:

((StateType(s49, ClassOfTradeCode) ∧ Value(s49, RS) ∧ In(g2, s49))
∧

(StateType(s410, PriceIdentifierCode) ∧ In(g2, s410) ∧ StateType(s411, RES) ∧ In(s410, s411)) ∧

(UnitPrice(s411, p22) ∧ Dimension(p22, USDollars) ∧ Unit(p22, count) ∧ Quantity(p22, 12.50)))

))

5.4 β_3: Assertions

In line 4 of segments 14 and 16, the buyer asserts that the basis of the unit price for the goods in question was a quote to this effect from the seller. Presumably, this is helpful information for processing the po.

There is more than one way legitimately to model this. Perhaps the simplest is to build upon a simple statement. Let *p1* be the price (of the stuff) in question. Then

EventType(e111, quote) ∧ SUBJ(e111, 'Fashion Today') ∧ DIR(e111, p1)
or
"Fashion Today quoted *p1*"

Since the po gives no temporal information, we leave this out, and since *p1* here is a name that has been given meaning elsewhere, we know (i.e., it follows) that it is a certain price for some certain goods.

Here then are the assertions in this po:

⊤ → (V(005001234500) ↔ (

(EventType(e111, quote) ∧ SUBJ(e111, 'Fashion Today') ∧ DIR(e111, p1)) ∧

(EventType(e112, quote) ∧ SUBJ(e112, 'Fashion Today') ∧ DIR(e111, p22))

))

5.5 β_4: promises

Although it is not explicit in the po, the buyer is promising to pay for the requested shipment. The promise, however, is conditioned, just as is the cancelation.

The condition on the promise is a subtle matter. The buyer is requesting shipment by 19980915 and declaring the po canceled if not shipped (or arrived?) by 19981015. What if the shipment occurs and the goods arrive before 19981015 and after 19980915? Perhaps the buyer is reserving the right to accept or reject the shipment. More complex situations can be modeled, of course. Here we'll keep it simple: the buyer promises to pay the seller for the goods on condition that the shipment request is honored.

H(005001234500, 1) → (K(005001234500) ↔ (

EventType(e61, pay) ∧ SUBJ(e61, 'The Corner Store') ∧ DIR(e61, 'Fashion Today') ∧ INDIR(e61, [g1, g2])))

References

File: span-fullEDI-sok-sam.tex. Created: 1999-11-16. Revision dates: 2000-07-08, 2000-8-22, 2000-8-23, 2000-8-25, 2000-8-26, 2001-02-24.

/* $Header$: $span - fullEDI - sok - sam.tex, v1.42000/08/2618$: 10 : $27sokExp$ */