Promise Theory, a Tool for System Specification

Jan A. Bergstra

Minstroom Research BV Utrecht,
&
Informatics Institute, Faculty of Science
University of Amsterdam
janaldertb@gmail.com@uva.nl

June 15, 2018

About myself

- PhD in mathematics in Utrecht in 1976.
- Since then in theory of computing (Utrecht University, Leiden University, Philips Research, University of Swansea, University of Amsterdam)
  - abstract data types,
  - Process algebra (ACP),
  - module algebra,
  - term rewriting,
  - instruction sequences
- Recent years:
  - complexity theory for instruction sequences,
  - theory of meadows (application to the theory of subjective probability),
  - informational money (Bitcoin etc.),
  - Promise Theory.

Origins of the work on PT

Around 2000 Mark Burgess from Oslo was in need of a specification language for his system CFengine (a version management system for systems software).

In 2003 he coined Promise Theory as a label for his approach.

In 2006 I joined Mark in the effort to develop promise theory.
Original Motivation for Mark Burgess:

1. Not satisfied with logic as a specification format.
2. Interested in providing a gradual transition from informal specs to formal specs.
3. Looking for more expressiveness than mere propositions which are true or false.
4. Also: keen on a notation admitting some formal or mathematical flavour.

Promises I

A promise is a speech action (communication, message) with some additional structure:

1. promiser X (an agent),
2. body B, (specifies what is promised),
3. promisee Y (an agent),
4. scope S (collection of agents who take notice of the promise).

$$X \rightarrow_S^B Y$$

B may be an action, or a state of affairs (current, past, or future).

Promise example I

For instance

1. promiser *db* (a database),
2. body "*p* has phone number 123454321",
3. promisee *u* (a database user),
4. scope $\{p\}$ (*p* is notified that *db* issues a promise about them.)

Now:

- It is immaterial whether or not the body is "true"!
- Databases issue (make) promises without any notion of truth.

Promise example II

For instance

1. promiser X (a person),
2. body "tomorrow I will bring you by car to the airport, leaving 12.00 from your hotel",
3. promisee Y (another person),
4. scope $\{Y_1, Y_2, Y_3\}$. (Friends of Y who stay in the same hotel.)

Now:

- It is immaterial whether or not X intends to perform as promised.
- There is no obligation for X to act as promised.

Keeping a promise

Assume

$$X \rightarrow^{B}_{\{U,V\}} Y$$

Now X is keeping its promise (for $Z \in \{U, V, Y\}$),

1. B comes about (according to the assessment of Z)
2. irrespective of causality (it is easy to keep the promise that the sun will rise tomorrow).

Now:

- It is immaterial whether or not X intends to keep its promise, AND,
- there is no obligation for X to act as promised.

Deontic neutrality

Mark Burgess: central CLAIM / ASSUMPTION / HYPOTHESIS:

Issuing a promise does NOT introduce an obligation for the promiser to keep the promise.

WHY?

1. machines (artificial agents) are insensitive to obligations,
2. conflicting obligations are highly problematic, (paradoxes of deontic logic),
3. all agents are autonomous,
4. autonomous agents don't impose obligations on one-another.

Effects of a promise I

So what is the effect of a promise $X \rightarrow^{B}_{\{U,V\}} Y$?

- State transition for promisee and for agents in scope: update of $P_Z(B)$ (the subjective probability for Z that E will come about)
- Creation of a life-cycle for promise body B for each agent in scope $((\{U, V\})$ and for the promisee.

Som quantification of these updates would be nice.

The effect of a promise II

Suppose

$$X \rightarrow^{B}_{\{U,V\}} Y$$

THEN

1. Y, U, X are likely to increase their assignment of subjective probability that B will come about.
2. the higher the trust in X the more the expectation that B will come about grows,
3. Once (or perhaps more often) Y makes an assessment of X's keeping of the promise,
4. If Y concludes that X is not keeping the promise the trust of X in Y decreases.
5. If U or V happens to notice that X is not keeping the promise (to Y) the trust of U or V in X may also decrease though less to than for Y.

Effects of a promise III (quantification)

Let $T_Z(X) \in [0,1]$ denote the trust of Z in X.
Upon receiving the promise from X: update the subjective probabilities $P_Z(B)$ for $Z \in \{Y\} \cup S$.
$P_Z(B) \rightarrow T_Z(X)$.

If the promise is kept (as checked at the end of the life-cycle): trust update for X (the less probable Y considers B the more impact it has if X keeps the promise):

$$T_Z(X) \rightarrow T_Z(X) + (1 - T_Z(X) \cdot (1 - P_Z(B)).$$

If the promise is not kept (according to Z):

$$T_Z(X) \rightarrow T_Z(X) \cdot (1 - P_Z(B))$$

The effect of a promise

Suppose

$$X \rightarrow^{B} Y$$

And suppose that promisee Y moderately trusts X. THEN

1. The main incentive for X to keep its promise (that is see to it that B comes about if X can do so) is that Y and the agents in scope will end up with higher trust in X and will be more likely to expect X to keep subsequent promises, a fact which may be helpful for X.
2. By repeatedly failing to keep its promises towards the samen agent Y, the trust of Y in X will gradually decrease.

Suppose also

$$X \rightarrow^{C} Z$$

and suppose that X cannot achieve both B and C at the same time.
THEN

- X has made conflicting promises.
- X may choose (B or C) what suits X best.

Conflicting promises: a matter of optimisation

Suppose

$$X \rightarrow^{B} Y$$

as well as

$$X \rightarrow^{C} Z$$

and suppose that X cannot achieve both B and C at the same time.

THEN

X may keep the promise issued to the agent whose trust is most important to X. A mere matter of optimisation.

Offering a service, accepting a service

Syntax with more detail.

X promises to offer service b to Y

$$X \to^{+b} Y$$

Y promises to accept service b from X

$$X \to^{-b} Z$$

In the presence of a collection of agents in scope S:

$$X \to_U^{+b} Y$$

and

$$X \to_V^{-b} Z$$

Protocols

Say first

$$X \to^{+b} Y$$

and then

$$X \to^{-b} Z$$

Now X and Y have agreed that X will provide service b to Y.

And with agents in scope:

$$X \to_U^{+b} Y$$

and

$$X \to_V^{-b} Z$$

Agents in $U \cap V$ are aware of the agreement.

Conditional promises

Promising B upon condition C becoming true,

$$C \Rightarrow X \rightarrow^{B} Y$$

With offer/accept notation:

$$C \Rightarrow X \rightarrow^{+b} Y$$

$$Y \rightarrow^{-b} X$$

For instance:
$C \equiv$ "Y clicks on a certain link", and
$+b \equiv$ "a certain webpage is served by X to Y".
$-b \equiv$ "use of the webpage is agreed by Y"

Indirect promises (promises of 2nd kind)

X promises Y that it will offer service b to V

$$X \to_S^{+b} Y[V]$$

X promises Y that Z will offer service b to V

$$X[Z] \to_S^{+b} Y[V]$$

X promises Y that Z will offer service b to V under condition C

$$C \Rightarrow X[Z] \to_S^{+b} Y[V]$$

Z is autonomous; X cannot impose anything on Z, thus X needs the promise:

$$requestBy(X) \Rightarrow Z \to_{S \cup \{Y\}}^{+b} X[V]$$

Now (being in scope of the promise by Z to X) Y may trust that X can keep its promise.

Promise types

Initial idea: promise is a documented intention (with additional features).

- documented intention: +, (promise to act)
- acceptance of documented intention: -,
- rejection of documented intention: -!,
- documented expectation: [+], (promise of fact) for instance:
  - the sun will rise tomorrow
  - tomorrow the weather will be fine
- agreement with documented expectation: [-], (agreement with promise of fact)
  - tomorrow's weather will indeed be ok
- rejection of documented expectation: [-!]
  - the weather tomorrow will not be good

Structured notation, further attributes

promise label P (serves as a reference)

promiser X

body B

promisee Y

condition C

scope S

event E (characteristic event at which promise is issued)

life cycle L (description of phases until promise is kept or forgotten)

bias b (background intentions of promiser)

type t (taken from $\{+, -, -!, [+], [-], [-!]\}$)

PT: alternative facts on the same level as facts

1. Promise theory (PT): facts understood as "promises of fact".
2. Alternative facts: just as well promises of fact.
3. PT: makes no distinction between facts and alternative facts.
4. PT: agents create and manipulate trust rather than truth, even more than promising "true" (non-alternative) facts.
5. Promising alternative facts may create trust.

Promises in a political context (including policy about IT):

1. Promising alternative facts is always an option.
2. No alternative fact is outdated forever, whatever information science has acquired about it.
3. Politics is uncommitted to truth.
4. Instead politics is committed to trust.

Threats

Promise theory may be extended with threats. There is no standard theory of threats so it seems. Threats play a role in IT just as well as in politics or in trade. Threats have specific properties:

- always conditional.
- body is harmful for promisee (in the perception of promisee and most agents in scope).
- the probability that the threat is not kept is significant (say above 1/3).

Promises and threats can be understood at various levels of abstraction, according to which features are taken into account.

FINALLY: book with MB (some copies available here), AND
Case Study on promises in the context of Brexit (some copies available).
I thank you for your attention!