Reasoning about Goals - part 1-

Agent-Oriented Software Engineering

A.A. 2019-2020

Prof. Paolo Giorgini

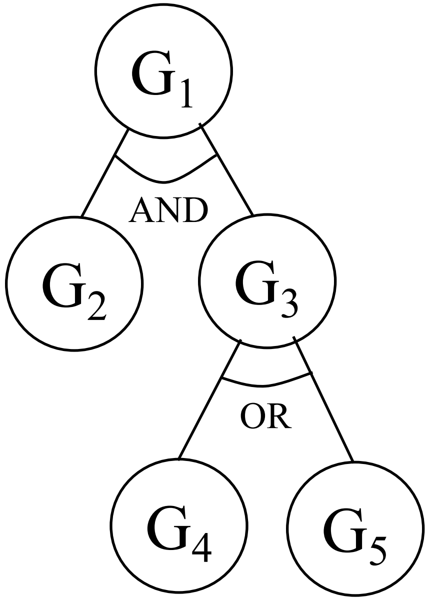
UNIVERSITY OF TRENTO - Italy

Department of Information and Communication Technology

Goal Analysis

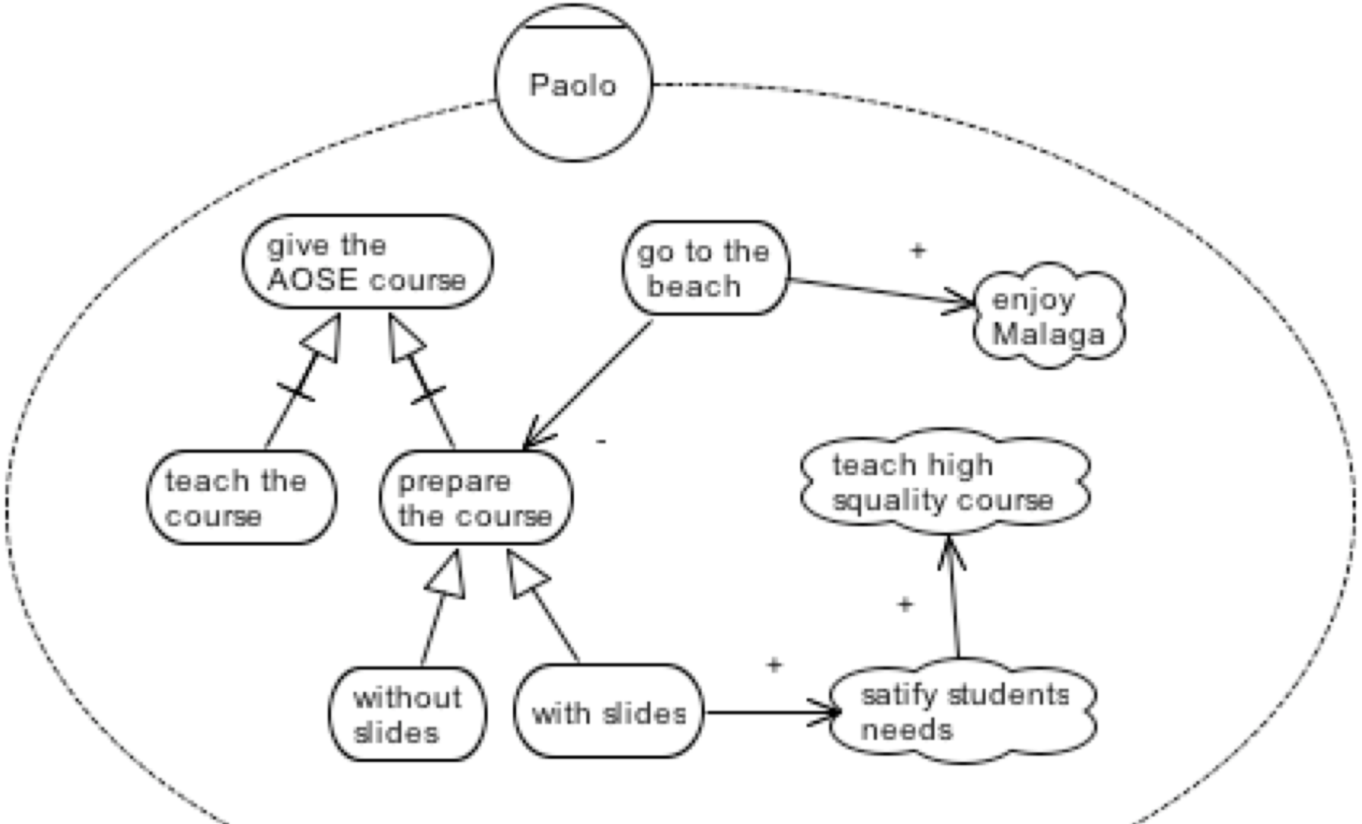
- Traditionally, goal analysis consists of decomposing goals into subgoals through an AND- or OR-decomposition.

  - Given a goal model and a set of initial labels for some goals (S for Satisfied and D for Denied) there is a simple labels propagation algorithm which can generate labels for all other goals of the model [Nilsson' 72]

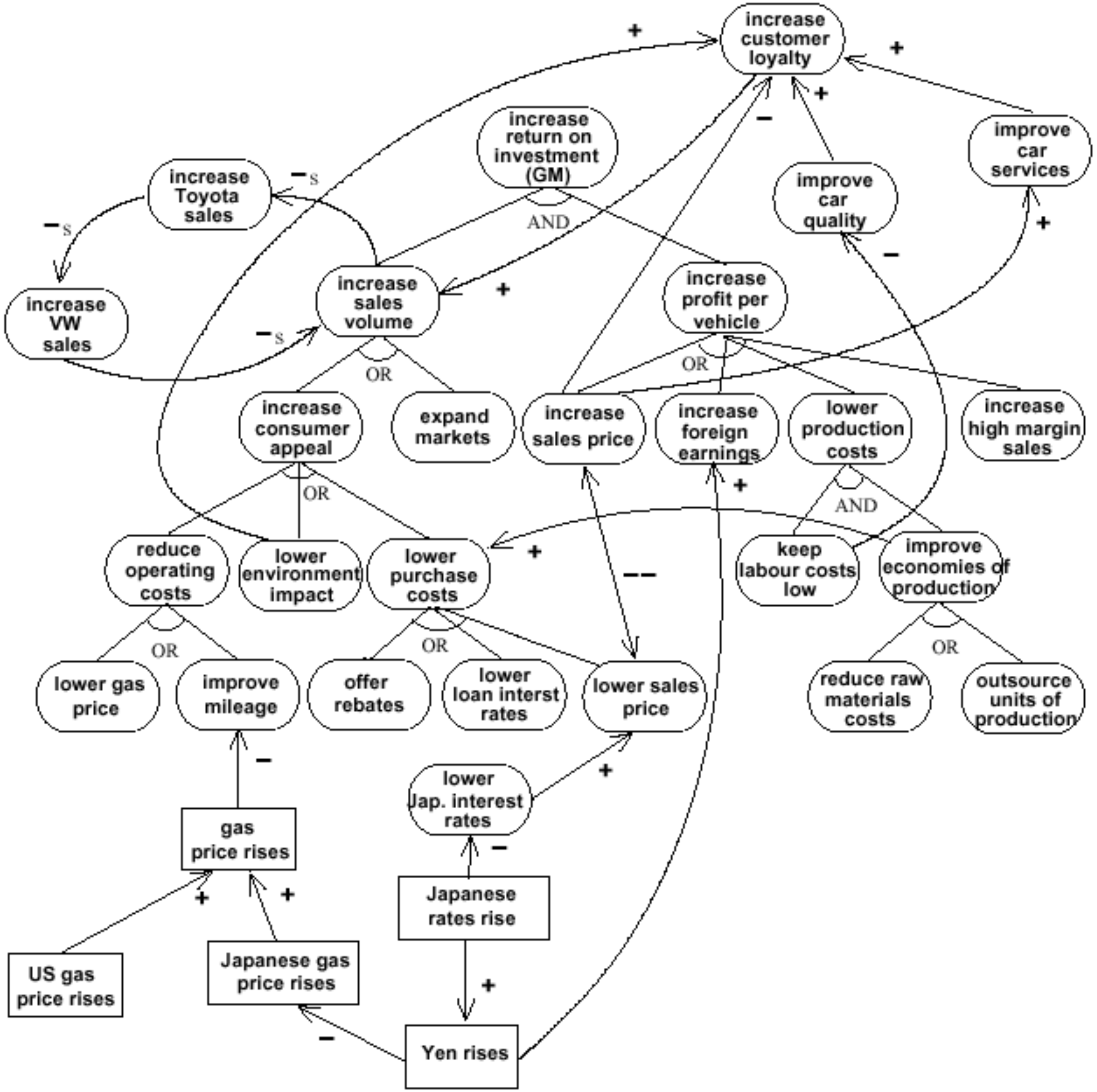


A simple example of goal model

- AND/OR decompositions
- Positive (+)/ Negative (-) contribution links



Part of a goal model for General Motor



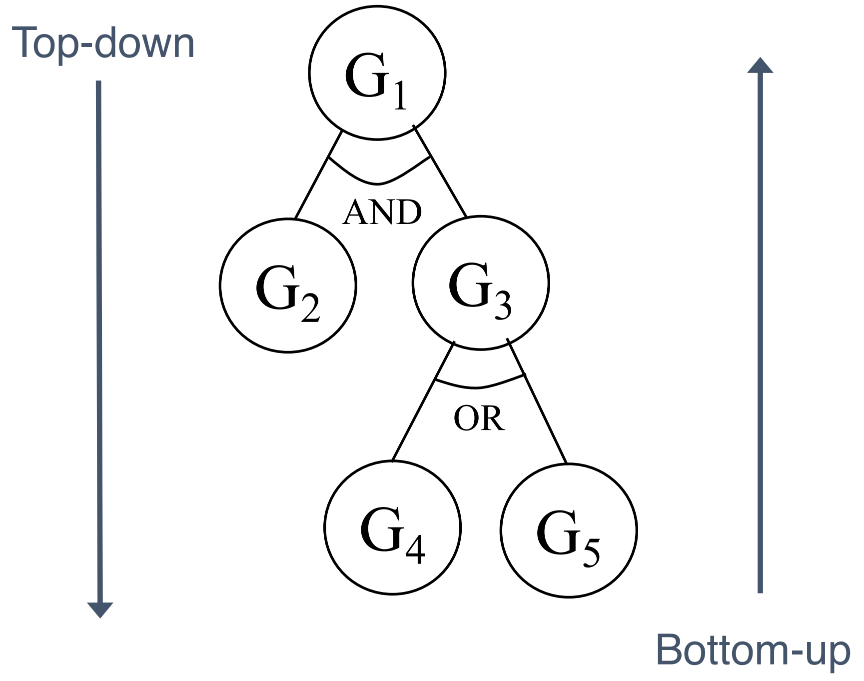
Goal Models

- Goal Dependency Graph:
  - Goals represented as Nodes
  - And/or relationships as (grouped) and/or arcs
  - Identity/negation as ++/- - arcs
  - Positive/negative contribution as +/- arcs
  - Cycles possible!

- Goal Valuations:
  - Goals can be either satisfied or denied
    - need to represent evidence of satisfaction/denial
  - Relationships propagate satisfaction and denial values
  - Conflicts possible!

The problem

- Formal representation(s) of goal models and goal valuations
    - Qualitative and quantitative approach
- Formal techniques to reason on goal values and on their propagation through goal models
    - Top-down (backward) reasoning
    - Bottom-up (forward) reasoning



Qualitative approach

- Four predicates:
  - FS(g): there is at least Full evidence that g is Satisfied
  - PS(g): there is at least Partial evidence that g is Satisfied
  - FD(g): there is at least Full evidence that g is Denied
  - PD(g): there is at least Partial evidence that g is Denied
- Negated atoms $\neg$FS(g), $\neg$FD(g) not admitted!
- FS(g)/PS(g) independent from FD(g)/PD(g)
  - This allow to have conflicts
    - E.g., *g* can be fully satisfied and partially denied
    - Different sources of information can provide both evidence for satisfaction and denial
    - A goal can receive a negative contribution from another goal (you cannot do both!) but its actual decomposition allow to satisfy the goal

Axiomatization

- Axioms allow to capture (define) the semantics of goal models
  - Express the semantics of relations and value propagation
  - Used to build sound reasoning techniques

Goal	Invariant Axioms
g	$FS(g) \rightarrow PS(g)$
	$FD(g) \rightarrow PD(g)$

Axiomatization

Goal Relation	Relation Axioms
$(G2,G3) \xrightarrow{AND} G1$:	$(FS(G2) \wedge FS(G3)) \rightarrow FS(G1)$
	$(PS(G2) \wedge PS(G3)) \rightarrow PS(G1)$
	$FD(G2) \rightarrow FD(G1)$, $FD(G3) \rightarrow FD(G1)$
	$PD(G2) \rightarrow PD(G1)$, $PD(G3) \rightarrow PD(G1)$
$(G2,G3) \xrightarrow{OR} G1$:	$(FD(G2) \wedge FD(G3)) \rightarrow FD(G1)$
	$(PD(G2) \wedge PD(G3)) \rightarrow PD(G1)$
	$FS(G2) \rightarrow FS(G1)$, $FS(G3) \rightarrow FS(G1)$
	$PS(G2) \rightarrow PS(G1)$, $PS(G3) \rightarrow PS(G1)$

Axiomatization

Goal Relation	Relation Axioms
$G2 \xrightarrow{++S} G1$:	$FS(G2) \rightarrow FS(G1)$, $PS(G2) \rightarrow PS(G1)$
$G2 \xrightarrow{--S} G1$:	$FS(G2) \rightarrow FD(G1)$, $PS(G2) \rightarrow PD(G1)$
$G2 \xrightarrow{+S} G1$:	$FS(G2) \rightarrow PS(G1)$, $PS(G2) \rightarrow PS(G1)$
$G2 \xrightarrow{-S} G1$:	$FS(G2) \rightarrow PD(G1)$, $PS(G2) \rightarrow PD(G1)$
$G2 \xrightarrow{++D} G1$:	$FD(G2) \rightarrow FD(G1)$, $PD(G2) \rightarrow PD(G1)$
$G2 \xrightarrow{--D} G1$:	$FD(G2) \rightarrow FS(G1)$, $PD(G2) \rightarrow PS(G1)$
$G2 \xrightarrow{+D} G1$:	$FD(G2) \rightarrow PD(G1)$, $PD(G2) \rightarrow PD(G1)$
$G2 \xrightarrow{-D} G1$:	$FD(G2) \rightarrow PS(G1)$, $PD(G2) \rightarrow PS(G1)$

Axiomatization (cont.)

- or, +D, -D, ++D, --D are dual w.r.t. and, +S, -S, ++S, --S
- Propagation of satisfaction through a ++, --, +, - may be or may be not symmetric w.r.t. that of denial:

$$G2 \xrightarrow{+} G1 \Leftrightarrow G2 \xrightarrow[S]{+} G1 \text{ and } G2 \xrightarrow{+D} G1$$

$$G2 \xrightarrow{-} G1 \Leftrightarrow G2 \xrightarrow{-S} G1 \text{ and } G2 \xrightarrow{-D} G1$$

Satisfaction/Dianial:

g is totally satisfied [resp. partially satisfied, totally/partially denied] iff FS(g) [resp. PS(g), FD(g), PD(g)] can be logically inferred from the initial assignment and the axioms

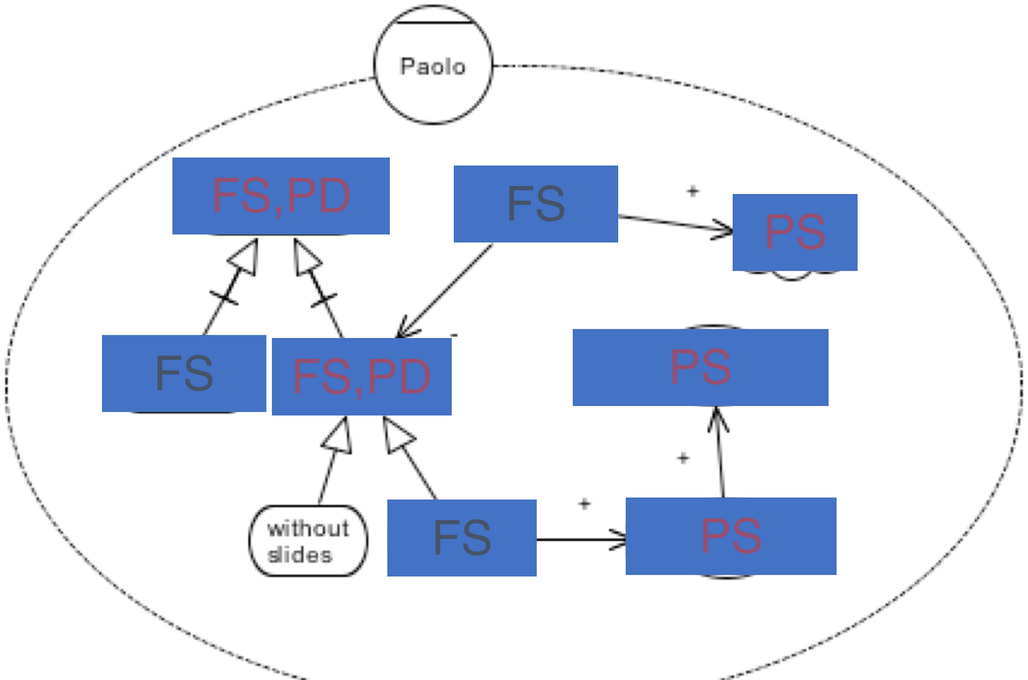
Forward Reasoning

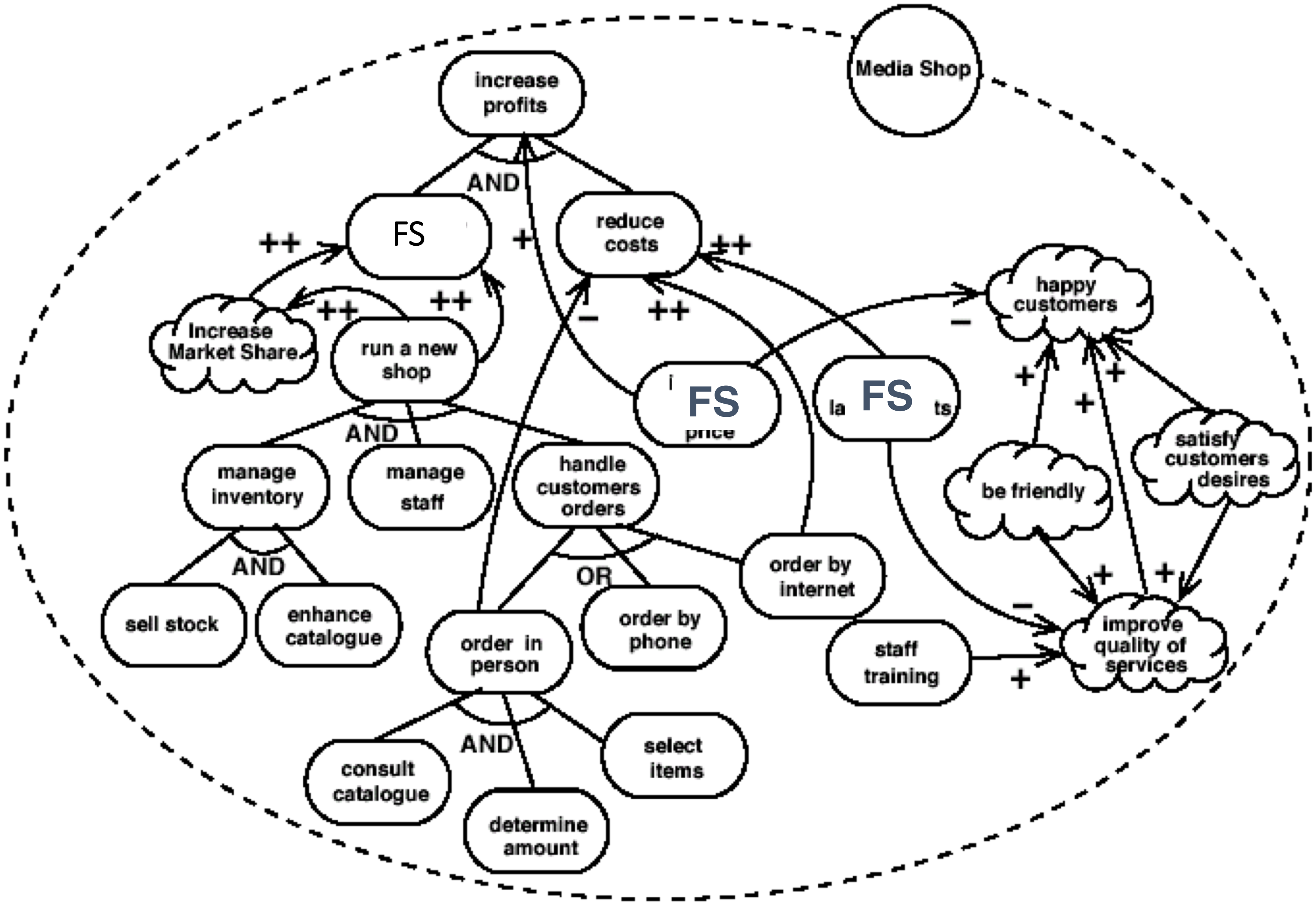
- Given
  - goal model
  - initial values assignment to some goals (input goals -- typically leaf goals)
- Forward reasoning focuses on the forward propagation of these initial values to all other goals of the graph accordingly to the axioms
- Initial values represents the evidence (possibly contradictory) available about the satisfaction and the denial of goals: {FS(G1), PD(G2), …}
  - Usually provided by the domain expert(s)

An example of Forward prop.

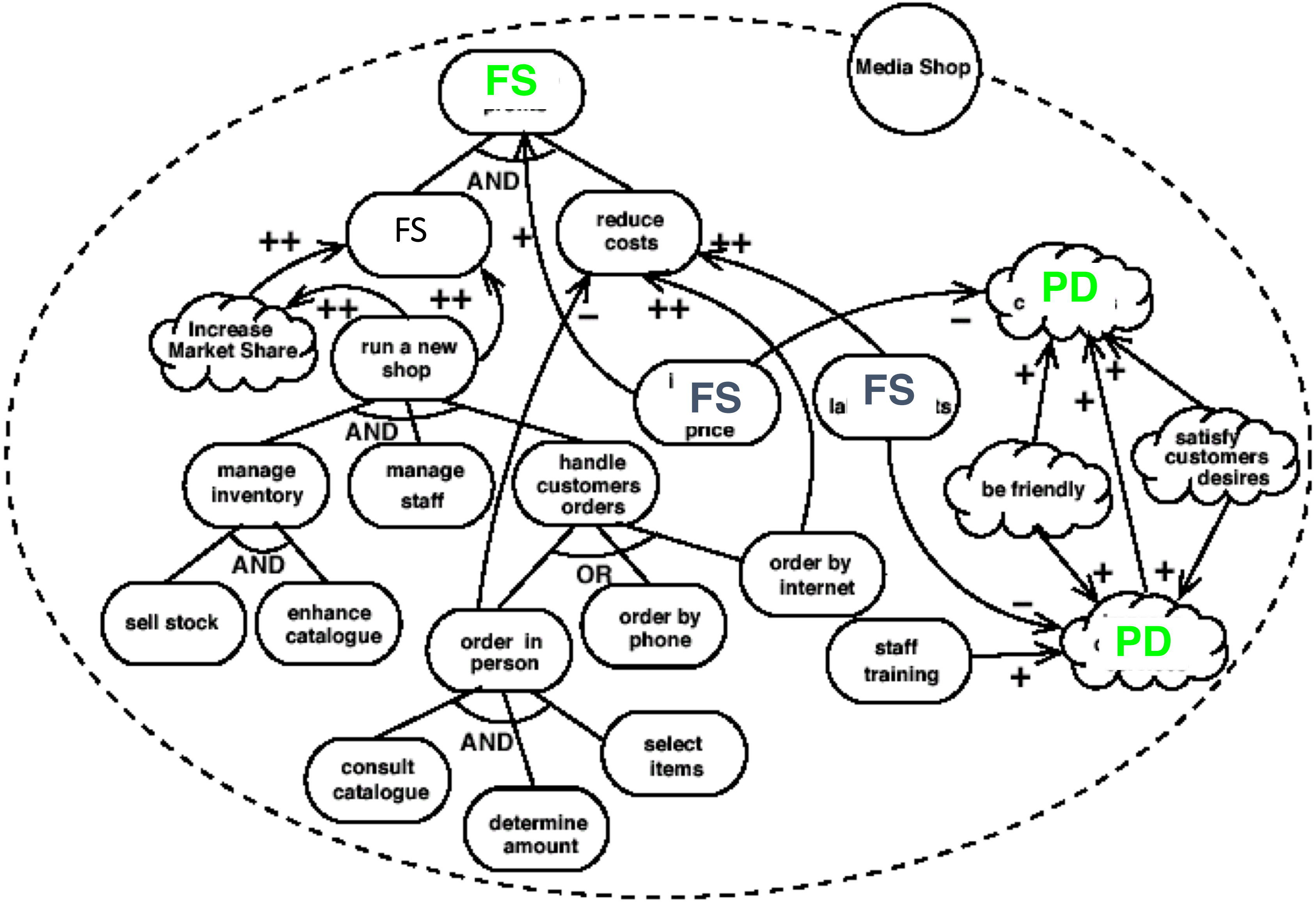
Initial: {FS(*teach the course*), FS(*with slides*), FS(*go to the beach*)}

Final : {FS(*teach the course*), FS(*with slides*), FS(*go to the beach*), FS(*prepare the course*), PD(*prepare the course*), PS(*enjoy Malaga*),PS(*satisfy students needs*), PS(*teach high quality course*)….}





Media Shop
increase profits
AND
FS
reduce costs
++
+
++
Increase Market Share
++
++
run a new shop
–
++
happy customers
–
FS
price
la
FS
ts
+
+
+
AND
satisfy customers desires
be friendly
manage inventory
manage staff
handle customers orders
order by internet
AND
OR
+
+
–
sell stock
enhance catalogue
order in person
order by phone
improve quality of services
staff training
+
AND
consult catalogue
select items
determine amount



Media Shop
FS
AND
FS
reduce costs
++
+
++
Increase Market Share
++
run a new shop
++
–
++
PD
–
FS
price
FS
AND
manage inventory
manage staff
handle customers orders
+
+
+
be friendly
satisfy customers desires
AND
OR
sell stock
enhance catalogue
order in person
order by phone
order by internet
+
+
–
staff training
+
PD
AND
consult catalogue
determine amount
select items

Propagation Algorithm

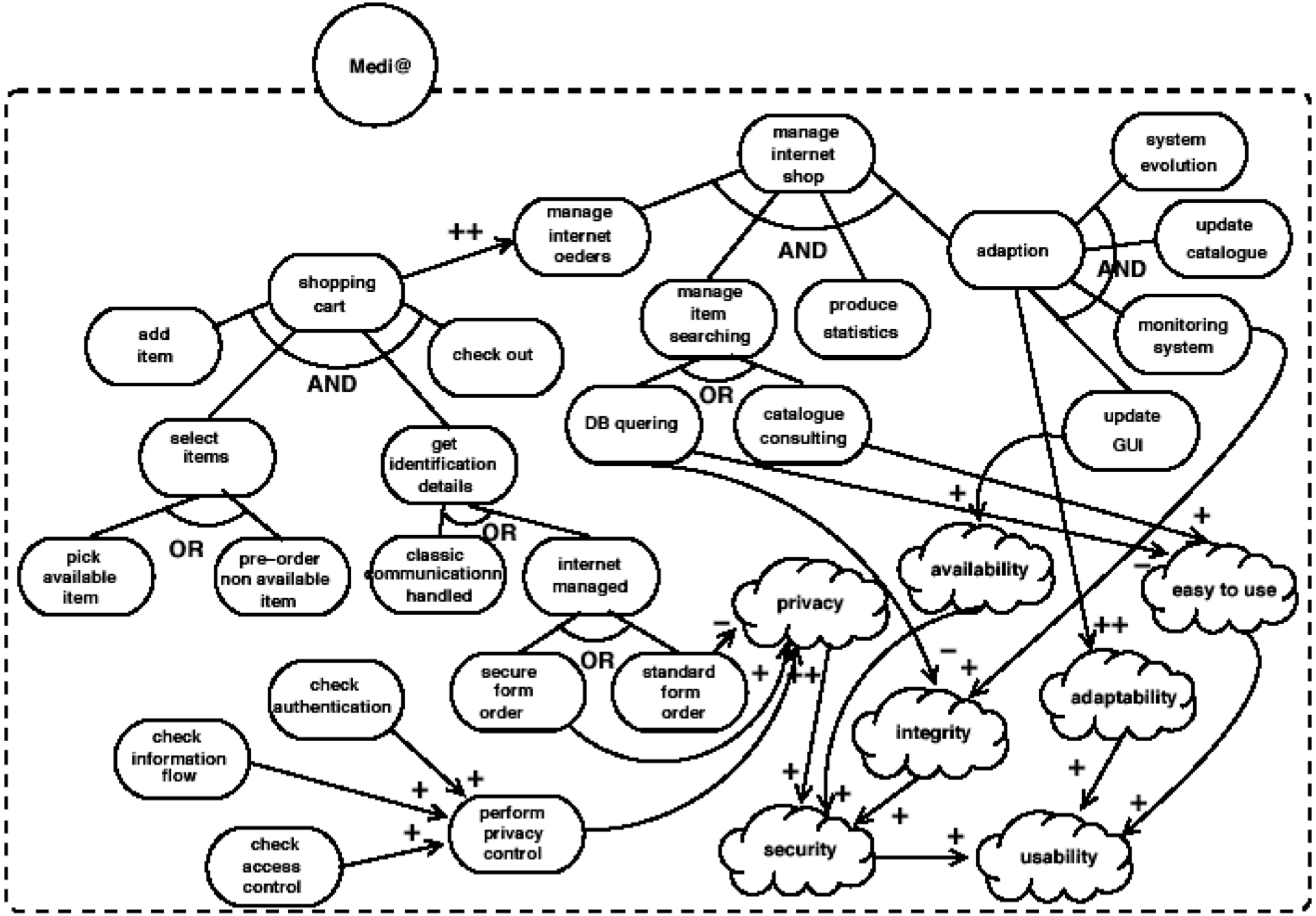
```
label_array Label_Graph(graph ⟨G,R⟩,label_array Initail)
  Current=Initial;
  do
    Old=Current;
     for each Gi ∈ G do
        Current[i]=Update_label(i,⟨G,R⟩,Old);
  until not (Current==Old);
  return Current;

label Update_label(int i, graph ⟨G,R⟩,label_array Old)
   for each Rj ∈ R  s.t. target(Ri)== Gi do
      satij = Apply_Rules_Sat(Gi,Rj,Old)
      denij = Apply_Rules_Den(Gi,Rj,Old)
    return ⟨max(maxj(satij),Old[i].sat),
                max(maxj(denij),Old[i].den)⟩
```

Propagation Algorithm (cont.)

- To each g we associate two variables Sat(g) and Den(g) ranging in {F,P,N} such that F>P>N
- E.g., Sat(g)≥P states that there is at least partial evidence that g is satisfiable
- From the initial assignment, we propagate the values according to the following rules:

	$(G_2, G_3) \overset{and}{\longmapsto} G_1$	$G_2 \overset{+}{\longmapsto} G_1$	$G_2 \overset{-}{\longmapsto} G_1$	$G_2 \overset{++}{\longmapsto} G_1$	$G_2 \overset{--}{\longmapsto} G_1$
$Sat(G_1)$	$min\{Sat(G_2), Sat(G_3)\}$	$min\{Sat(G_2), P\}$	N	$Sat(G_2)$	N
$Den(G_1)$	$max\{Sat(G_2), Sat(G_3)\}$	N	$min\{Sat(G_2), P\}$	N	$Sat(G_2)$

- or, +D, -D, ++D, --D dual w.r.t. and, +S, -S, ++S, --S
- Satisfaction/denial values monotonically non-decreasing
- Terminates when reaches a fixpoint (*Current==Old*)


Medi@
manage internet shop
manage internet oeders
shopping cart
add item
check out
select items
get identification details
pick available item
pre–order non available item
classic communicationn handled
internet managed
secure form order
standard form order
check authentication
check information flow
check access control
perform privacy control
manage item searching
produce statistics
DB quering
catalogue consulting
adaption
system evolution
update catalogue
monitoring system
update GUI
privacy
availability
easy to use
integrity
adaptability
security
usability
AND
OR
++
+
–

Goals	Exp 1				Exp 2				Exp 3				Exp 4			
	Init		Fin		Init		Fin		Init		Fin		Init		Fin	
	S	D	S	D	S	D	S	D	S	D	S	D	S	D	S	D
DB querying	F		F		F		F									
catalogue consulting									F		F		F		F	
pick available item	F		F		F		F		F		F		F		F	
pre-order non available item																
classic communication handled	F		F													
standard form order					F		F		F		F					
secure form order													F		F	
manage internet shop			F				F				F				F	
privacy			P				P	P			P	P			P	
availability			P				P				P				P	
integrity			P	P			P				P	P			P	
usability			P				P				P				P	
adaptability			F				F				F				F	
easy to use				P				P			P				P	
security			P				P				P				P	

Reasoning about Goals
- Part 2-

Agent-Oriented Software Engineering

A.A. 2019-2020

Prof. Paolo Giorgini

UNIVERSITY OF TRENTO - Italy

Department of Information
and Communication Technology

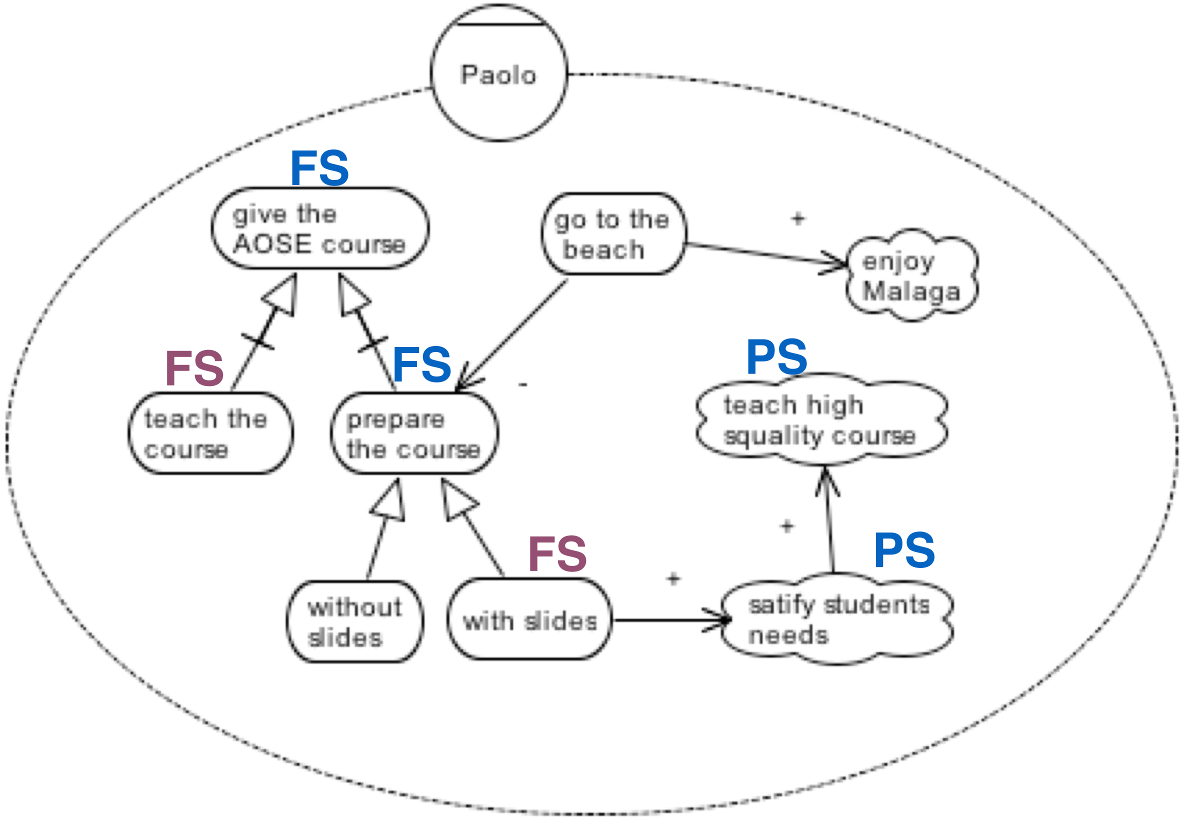
Backward reasoning

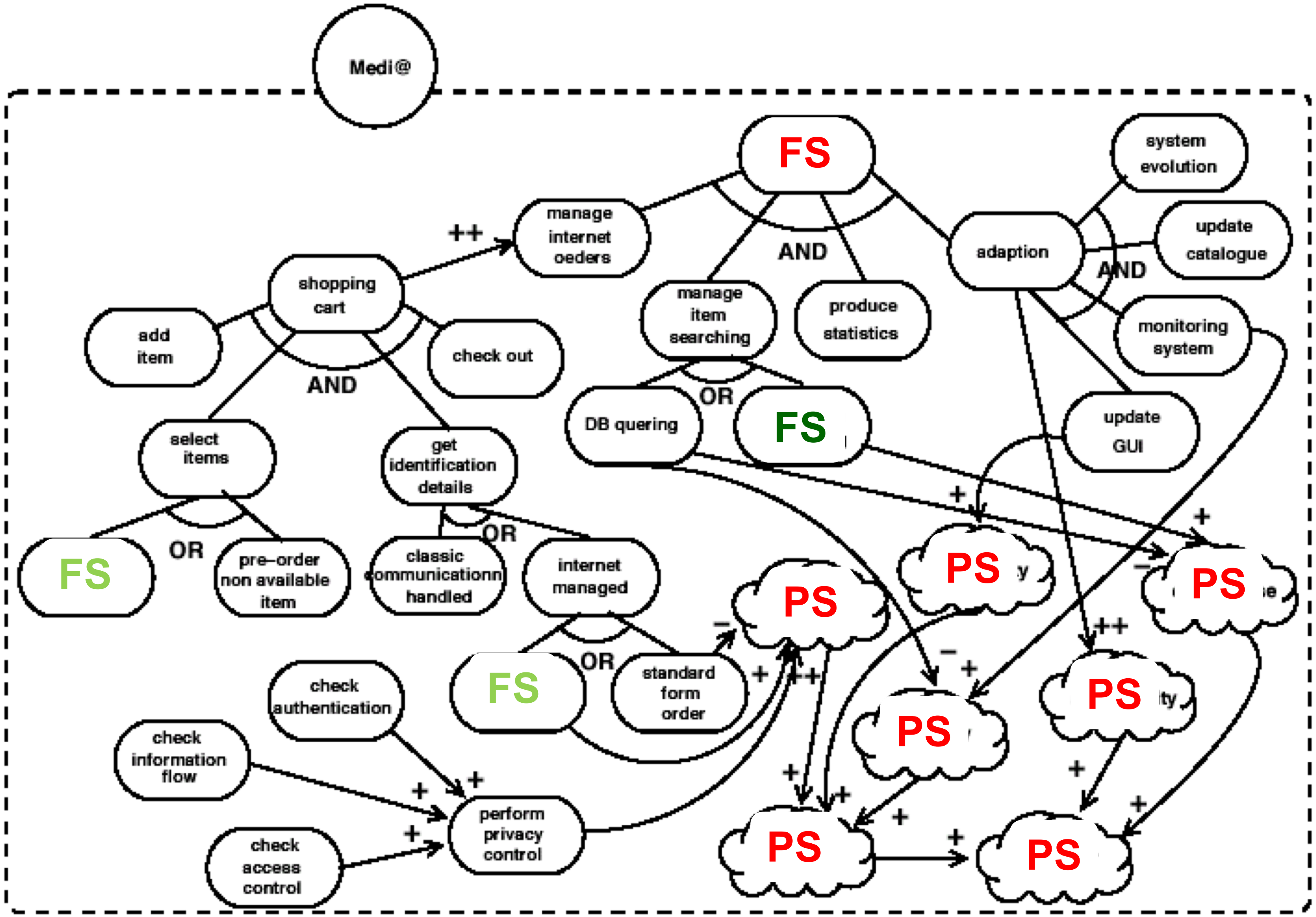
- We set the desired final values of the target goals, and we want to find possible initial assignments to the input goals which would cause the desired final values of the target goals
- We search for possible initial assignments to the input goals which would cause the desired final values of the target goals by forward propagation
- The user may also add some desired constraints, and decide to avoid conflicts

An example of Backward prop.

Final: {FS(*give AOSE course*), PS(*teach high quality course*)}

Assign. : {FS(*teach the course*), FS(*with slides*)}




Medi@
FS
manage internet oeders
++
shopping cart
AND
add item
check out
select items
get identification details
OR
FS
pre-order non available item
classic communicationn handled
internet managed
OR
FS
standard form order
check authentication
check information flow
perform privacy control
check access control
manage item searching
produce statistics
OR
DB quering
FS
adaption
AND
system evolution
update catalogue
monitoring system
update GUI
PS
PS
PS
PS
PS
PS
PS

Constraints and Costs

- We may also add some desired constraints and decide to avoid
  - Strong conflict (e.g., FS(G),FD(G))
  - Medium conflict (e.g., FS(G),PD(G))
  - Weak (e.g., PS(G),PD(G))
  - all conflicts
- Assigning a cost to each input goal, we search for an assignment at the minimum cost

Propositional Satisfiability (SAT)

- We reduce the backward search to a SAT problem
- SAT is the problem of determining whether a boolean formula Φ admits at least one satisfying truth assignment μ to its variables A_i
- SAT is a NP-complete problem (there does not exist any polynomial algorithm able to solve it)
- There exists efficient SAT techniques
  - DPLL is the most popular SAT algorithm
  - **CHAFF** is the most efficient DPLL implementation
- There are several techniques to improve the efficiency of DPLL (e.g., backjumping, learning, random restart)

Minimum-Weight SAT (MW-SAT)

- MW-SAT is a variant of SAT, where the boolean variables A_i occurring in Φ are given a positive integer weight w_i
- MW-SAT is the problem of determining a truth assignment μ satisfying Φ which minimizes the value

$$W(\mu) := \sum_i \{w_i \mid A_i \textit{ is assigned } \top \textit{ by } \mu\},$$

  or stating there is none.
- **MINWEIGHT** is the state-of-art solver for MW-SAT

Basic Formalization

- The boolean variables of Φ are all the values FS(G),PS(G),FD(G),PD(G) for each goal G and Φ is

$$\Phi := \Phi_{graph} \wedge \Phi_{outval} \wedge \Phi_{backward} \, [\wedge \, \Phi_{constraints} \wedge \Phi_{conflict}]$$

where

Φ_{graph} encodes the *goal graph*

Φ_{outval} encodes the *desired final output values*

$\Phi_{backward}$ encodes the *backward reasoning*

$\Phi_{constraints}$ encodes *user's constraints* (optional)

$\Phi_{conflict}$ encodes the *prevention of conflicts* (optional)

Basic Formalization cont.

- **Encoding the goal graph**

$$\Phi_{graph} := \bigwedge_{G \in \mathcal{G}} Invar\_Ax(G) \wedge \bigwedge_{r \in \mathcal{R}} Rel\_Ax(r)$$

Invar_Ax(G) is the conjunction of the invariant axioms and *Rel_Ax(r)* is the conjunction of the relation axioms (forward propagation through the relation arcs in the graph)

- **Representing Desired Final Output Values**

$$\Phi_{outval} := \bigwedge_{G \in Target(\mathcal{G})} vs(G) \wedge \bigwedge_{G \in Target(\mathcal{G})} vd(G)$$

Target($\mathcal{G}$) is the set of target goals and *vs(G)* $\in$ {T,PS(G),FS(G)}, *vd(G)* $\in$ {T,PD(G),FD(G)} are the maximum satisfiability and deniability values assigned to the target goal G.

Basic Formalization cont.

- **Econding Backward Reasoning**

$$\Phi_{backward} := \bigwedge_{G \in \mathcal{G}/Input(\mathcal{G})} \bigwedge_{v(G)} Backward\_Ax(v(G))$$

$$Backward\_Ax(v(G)) := v(G) \rightarrow \bigvee_{r \in Incoming(G)} Prereqs(v(G), r)$$

Input(𝒢) is the set of input goals; *Incoming(G)* is the set of relations in G; *v(G)={PS(G),FS(G),PD(G),FD(G)}*, and *Prereqs*(*v*(G),*r*) is a formula which is true iff the prerequisites of *v*(G) through *r* hold.

Backward_Ax(*v* (G)) is the set of propagation axioms (see next slide)

If G is not an input goal and *v*(G) holds, then this value must derive from the prerequisite values of some incoming relations of G

Axioms for backward propagation

$$
FS(G) \rightarrow \begin{cases}
\bigwedge_i FS(G_i) \ \vee & \text{If } (G_1, ..., G_i, ... G_n) \xmapsto{and} G \\
\bigvee_i FS(G_i) \ \vee & \text{If } (G_1, ..., G_i, ... G_n) \xmapsto{or} G \\
FS(G_i) \ \vee & \text{For every } R_i\text{: } G_i \xmapsto{++S} G \\
FD(G_i) & \text{For every } R_i\text{: } G_i \xmapsto{--D} G
\end{cases}
\qquad
FD(G) \rightarrow \begin{cases}
\bigvee_i FD(G_i) \ \vee & \text{If } (G_1, ..., G_i, ... G_n) \xmapsto{and} G \\
\bigwedge_i FD(G_i) \ \vee & \text{If } (G_1, ..., G_i, ... G_n) \xmapsto{or} G \\
FD(G_i) \ \vee & \text{For every } R_i\text{: } G_i \xmapsto{++D} G \\
FS(G_i) & \text{For every } R_i\text{: } G_i \xmapsto{--S} G
\end{cases}
$$

$$
PS(G) \rightarrow \begin{cases}
\bigwedge_i PS(G_i) \ \vee & \text{If } (G_1, ..., G_i, ... G_n) \xmapsto{and} G \\
\bigvee_i PS(G_i) \ \vee & \text{If } (G_1, ..., G_i, ... G_n) \xmapsto{or} G \\
PS(G_i) \ \vee & \text{For every } R_i\text{: } G_i \xmapsto{++S} G \\
PD(G_i) \ \vee & \text{For every } R_i\text{: } G_i \xmapsto{--D} G \\
PS(G_i) \ \vee & \text{For every } R_i\text{: } G_i \xmapsto{+S} G \\
PD(G_i) & \text{For every } R_i\text{: } G_i \xmapsto{-D} G
\end{cases}
\qquad
PD(G) \rightarrow \begin{cases}
\bigvee_i PD(G_i) \ \vee & \text{If } (G_1, ..., G_i, ... G_n) \xmapsto{and} G \\
\bigwedge_i PD(G_i) \ \vee & \text{If } (G_1, ..., G_i, ... G_n) \xmapsto{or} G \\
PD(G_i) \ \vee & \text{For every } R_i\text{: } G_i \xmapsto{++D} G \\
PS(G_i) \ \vee & \text{For every } R_i\text{: } G_i \xmapsto{--S} G \\
PD(G_i) \ \vee & \text{For every } R_i\text{: } G_i \xmapsto{+D} G \\
PS(G_i) & \text{For every } R_i\text{: } G_i \xmapsto{-S} G
\end{cases}
$$

Optional components

Adding User's Constraints and Desiderata

$$\Phi_{outval} := \bigwedge_i \bigvee_j lit_{ij};$$

The user expresses constraints and desiderata on goal values (e.g., "PS(G_1)" means "G_1 is at least partial satisfiable", but it might totally satisfiable

A negative clause value is used to prevent a value to a goal (e.g.,"$\neg$FD(G_1)" means "G_1 cannot be fully deniable", but it might be partially deniable)

FS(G_1)$\vee$FS(G_2) means at least G_1 or G_2 must be fully satisfiable

Basic Formalization cont.

Preventing conflicts

It allows the user for looking for solutions which do not involve conflicts

Strong conflicts

$$\Phi_{conflict} := \bigwedge_{G \in \mathcal{G}} (\neg FS(G) \vee \neg FD(G))$$

Strong and medium conflicts

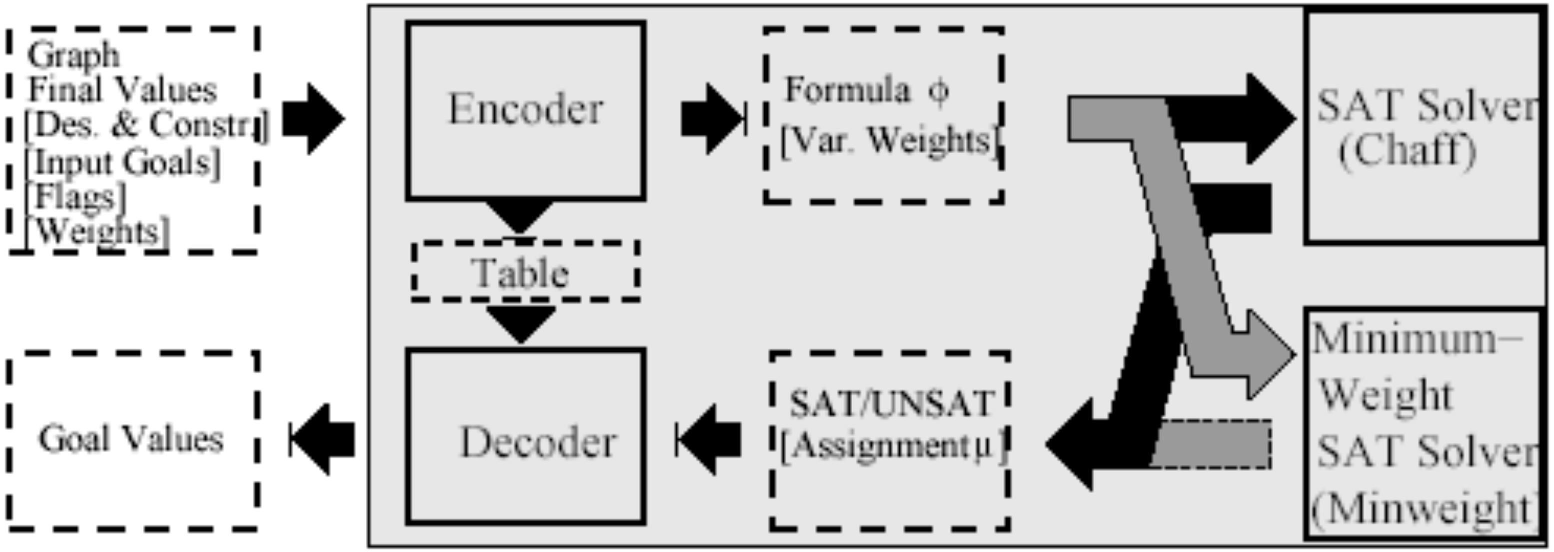
$$\Phi_{conflict} := \bigwedge_{G \in \mathcal{G}} ((\neg FS(G) \vee \neg PD(G)) \wedge (\neg PS(G) \vee \neg FD(G)$$

All conflicts

$$\Phi_{conflict} := \bigwedge_{G \in \mathcal{G}} (\neg PS(G) \vee \neg PD(G))$$

Backward Prop. implementation

- The qualitative backward propagation has been reduced to the Satisfiability (SAT) and minimum-cost satisfiability (minimum-cost SAT) problems for Boolean formulas
- GOLSOLVE / GOLMINSOLVE



	C=0				C=16				C=14				C=13			
Goals	Exp 1				Exp 2				Exp 3				Exp 4			
	Init		Fin		Init		Fin		Init		Fin		Init		Fin	
	S	D	S	D	S	D	S	D	S	D	S	D	S	D	S	D
DB querying (3)															F	
catalogue consulting (6)							F				F					
pick available item (2)							F				F				F	
pre-order non available item (7)																
classic communication handled (4)																
standard form order (6)											F					
secure form order (8)							F								F	
manage internet shop	F				F		F		F		F		F		F	
privacy	F				P		P		P		P	P	P		P	
availability	F				P		P		P		P				P	
integrity	F				P		P		P		P	P			P	
usability	F				P		P		P		P				P	
adaptability	F				P		F		P		F				F	
easy to use	F				P		P		P		P					P
security	F				P		P		P		P				P	

Avoiding conflicts

Quantitative reasoning

Quantitative Approach

- Evidence of satisfaction/denial represented by real values in $\mathcal{D}$: $[inf, sup]$, $0 \leq inf < sup$
- Value propagation through goal graphs as math functions, $f : \mathcal{D}^n \rightarrow \mathcal{D}$
- Much finer-grained:
  - Different degrees of satisfaction/denial evidence
  - Different degrees of positive/negative contribution
  - Different strength of conflicts

Numerical representation of evidence

- Sat(g), Den(g) $\in [inf,sup]$
- Atoms in the form Sat(g) $\geq c_1$ [Den(g) $\geq c_2$]: "there is at least an evidence c_1 [c_2] that g is Satisfied [Denied]"

$$c_1 = inf,\ c_2 = inf \Leftrightarrow \top$$

$$c_1, c_2 \in]inf,sup[\Leftrightarrow PS(g), PD(g)$$

$$c_1 = sup, c_2 = sup \Leftrightarrow FS(g), FD(g)$$

- *Conflict: Sat(g)$\geq c_1$* and *Den(g)$\geq c_2$*, $c_1, c_2 \in]inf,sup]$

Value propagation model

- 2 dual OPERATORS: $\oplus$ and $\otimes$, representing value propagation trough "or" and "and"
  - Independent probability model:

    $inf=0,\ sup=1$

    $p_1 \oplus p_2 = p_1+p_2 - p_1 \cdot p_2$

    $p_1 \otimes p_2 = p_1 \cdot p_2$

    (disjunction and conjuction of two independent events of probability p_1 and p_2)
  - *Flow model* (Resistor):

    $inf=0,\ sup = +\infty$

    $v_1 \oplus v_2 = v_1+v_2$

    $v_1 \otimes v_2 = (v_1 \cdot v_2)/(v_1+v_2)$
  - …

Axiomatization

Goal Relation   Relation Axioms

$$(G2,G3) \xrightarrow{AND} G1: \quad (Sat(G2) \geq x \wedge Sat(G3) \geq y) \rightarrow Sat(G1) \geq (x \otimes y)$$
$$(Den(G2) \geq x \wedge Den(G3) \geq y) \rightarrow Den(G1) \geq (x \oplus y)$$

$$(G2,G3) \xrightarrow{OR} G1: \quad (Sat(G2) \geq x \vee Sat(G3) \geq y) \rightarrow Sat(G1) \geq (x \oplus y)$$
$$(Den(G2) \geq x \vee Den(G3) \geq y) \rightarrow Den(G1) \geq (x \otimes y)$$

- AND and OR relation are dual

Axiomatization cont.

Goal Relation Relation Axioms

$G2 \xrightarrow{w+S} G1:$	$Sat(G2) \geq x \rightarrow Sat(G1) \geq (x \otimes w)$
$G2 \xrightarrow{w-S} G1:$	$Sat(G2) \geq x \rightarrow Den(G1) \geq (x \otimes w)$
$G2 \xrightarrow{++S} G1:$	$Sat(G2) \geq x \rightarrow Sat(G1) \geq x$
$G2 \xrightarrow{--S} G1:$	$Sat(G2) \geq x \rightarrow Den(G1) \geq x$
$G2 \xrightarrow{w+D} G1:$	$Den(G2) \geq x \rightarrow Den(G1) \geq (x \otimes w)$
$G2 \xrightarrow{w-D} G1:$	$Den(G2) \geq x \rightarrow Sat(G1) \geq (x \otimes w)$
$G2 \xrightarrow{++D} G1:$	$Den(G2) \geq x \rightarrow Den(G1) \geq x$
$G2 \xrightarrow{--D} G1:$	$Den(G2) \geq x \rightarrow sat(G1) \geq x$

- +D, -D, ++D, --D dual w.r.t. +S, -S, ++S, --S
- Remark: + and - relations have a weight w

Quantitative propagation

- There is at least an evidence c that g is satisfied [resp. denied] iff $Sat(g) \geq c$ [resp. $Den(g) \geq c$] can be logically inferred from the initial assignment and the axioms.
- $Sat(g) \geq c$, $Den(g) \geq c$ propagated independently

Forward Propagation Algorithm

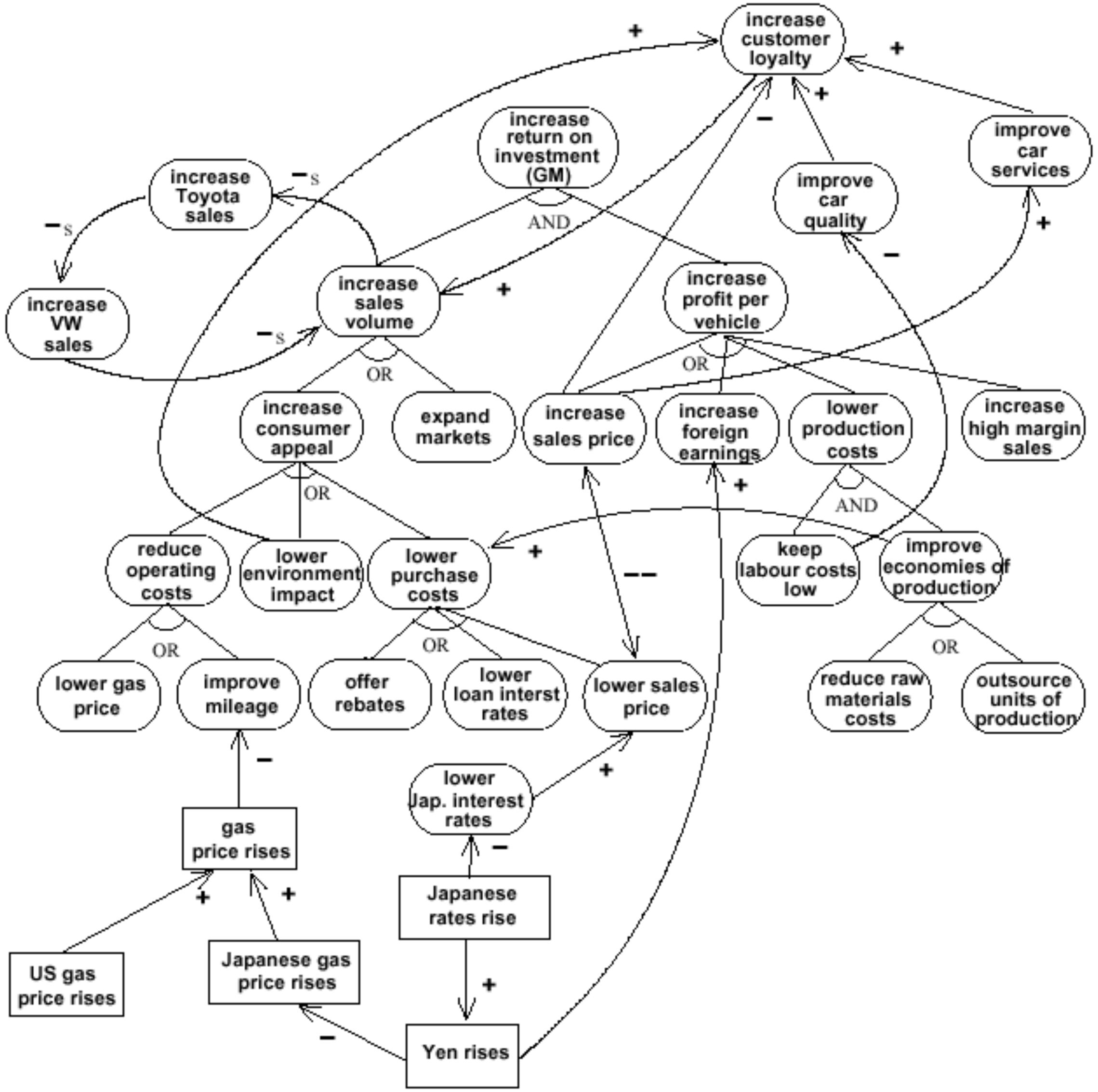
```
label_array Label_Graph(graph ⟨G,R⟩,label_array Initail)
   Current=Initial;
   do
      Old=Current;
       for each Gi ∈ G do
          Current[i]=Update_label(i,⟨G,R⟩,Old);
   until not (||Current – Old||∞ ≤ ε);
   return Current;

label Update_label(int i, graph ⟨G,R⟩,label_array Old)
    for each Rj ∈ R  s.t. target(Ri)== Gi do
       satij = Apply_Rules_Sat(Gi,Rj,Old)
       denij = Apply_Rules_Den(Gi,Rj,Old)
     return ⟨max(maxj(satij),Old[i].sat),
              max(maxj(denij),Old[i].den)⟩
```

Forward Propagation Algorithm

	$(G_2, G_3) \xmapsto{and} G_1$	$G_2 \xmapsto{w+} G_1$	$G_2 \xmapsto{w-} G_1$	$G_2 \xmapsto{++} G_1$	$G_2 \xmapsto{--} G_1$
$Sat(G_1)$	$Sat(G_2) \otimes Sat(G_3)$	$Sat(G_2) \otimes w$		$Sat(G_2)$	
$Den(G_1)$	$Den(G_2) \oplus Den(G_3)$		$Sat(G_2) \otimes w$		$Sat(G_2)$

- Satisfaction/denial values monotonically non-decreasing
- Uses Cauchy-convergence as termination condition:

$$|a_{n+1} - a_n| \xrightarrow{n \to \infty} 0$$


increase customer loyalty
increase return on investment (GM)
AND
increase Toyota sales
increase VW sales
increase sales volume
improve car quality
improve car services
increase profit per vehicle
OR
increase consumer appeal
expand markets
increase sales price
increase foreign earnings
lower production costs
increase high margin sales
OR
AND
reduce operating costs
lower environment impact
lower purchase costs
keep labour costs low
improve economies of production
OR
OR
OR
lower gas price
improve mileage
offer rebates
lower loan interst rates
lower sales price
reduce raw materials costs
outsource units of production
lower Jap. interest rates
gas price rises
Japanese rates rise
US gas price rises
Japanese gas price rises
Yen rises

Quantitative approach: example

Goal/Event	Relationship	Goal/Event
increase sales volume	$\xrightarrow{0.6-S}$	increase Toyota sales
increase Toyota sales	$\xrightarrow{0.6-S}$	increase VW sales
increase VW sales	$\xrightarrow{0.6-S}$	increase sales volume
increase customer loyalty	$\xrightarrow{0.4+}$	increase sales volume
increase sales prices	$\xrightarrow{0.5-}$	increase customer loyalty
increase car quality	$\xrightarrow{0.8+}$	increase customer loyalty
improve car services	$\xrightarrow{0.7+}$	increase customer loyalty
lower environment impact	$\xrightarrow{0.4+}$	increase customer loyalty
increase sales prices	$\xrightarrow{0.3+}$	improve car services
keep labour costs low	$\xrightarrow{0.7-}$	increase car quality
improve economies of production	$\xrightarrow{0.8+}$	lower purchase costs
Yen rises	$\xrightarrow{0.8+}$	increase foreign earnings
lower Japanese interest rates	$\xrightarrow{0.4+}$	lower sales price
Japanese rates rises	$\xrightarrow{0.8-}$	lower Japanese interest rates
Japanese rates rises	$\xrightarrow{0.6+}$	Yen rises
Yen rises	$\xrightarrow{0.4-}$	Japanese gas price rises
Japanese gas price rises	$\xrightarrow{0.6+}$	gas price rises
US gas price rises	$\xrightarrow{0.6+}$	gas price rises
gas price rises	$\xrightarrow{0.8-}$	improve mileage

Goals/Events	Exp 1				Exp 2				Exp 3					
	Init		Fin		Init		Fin		Init		Fin		Init	
	S	D	S	D	S	D	S	D	S	D	S	D	S	D
increase return on investment (GM)	0.0	0.0	0.0	0.4	0.0	0.0	0.8	0.4	0.0	0.0	0.9	0.0	0.0	0.0
increase sales volume	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
increase profit per vehicle	0.0	0.0	0.0	0.4	0.0	0.0	0.8	0.4	0.0	0.0	0.9	0.0	0.0	0.0
increase customer appeal	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
expand markets	0.3	0.0	0.3	0.0	0.3	0.0	0.3	0.0	0.3	0.0	0.3	0.0	0.3	0.0
increase sales price	0.0	0.5	0.0	0.8	0.0	0.5	0.0	0.8	0.0	0.5	0.0	0.8	0.0	0.5
increase foreign earnings	0.0	0.9	0.0	0.9	0.0	0.9	0.8	0.9	0.0	0.9	0.8	0.9	0.0	0.9
lower production costs	0.0	0.0	0.0	0.9	0.0	0.0	0.0	0.9	0.0	0.0	0.6	0.0	0.0	0.0
increase high margin sales	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6
reduce operating costs	0.0	0.0	0.8	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.8	0.0	0.0	0.0
lower environmental impact	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.0
lower purchase costs	0.0	0.0	0.9	0.0	0.0	0.0	0.9	0.0	0.0	0.0	0.9	0.0	0.0	0.0
keep labour costs low	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.9	0.9	0.0	0.9	0.0	0.9	0.0
improve economies of production	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0
lower gas price	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
improve mileage	0.8	0.0	0.8	0.0	0.8	0.0	0.8	0.0	0.8	0.0	0.8	0.0	0.8	0.0
offer rebates	0.3	0.0	0.3	0.0	0.3	0.0	0.3	0.0	0.3	0.0	0.3	0.0	0.3	0.0
lower loan interest rates	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
lower sales price	0.8	0.0	0.8	0.0	0.8	0.0	0.8	0.0	0.8	0.0	0.8	0.0	0.8	0.0
reduce raw materials costs	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.7	0.0	0.7	0.0
outsource units of production	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
gas price rises	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
lower Japanese interest rates	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
US gas price rises	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Japanese gas price rises	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.0	0.4	0.0	0.0
Yen rises	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0

Backward Propagation

- Formalization of the problem:
  - Linear cost function: min Ax
  - a set of non linear equality and inequality constraints
- ... and we pass the system to Lindo 8.0. www.lindo.com

G(non-input):

$$Sat(G) = max \begin{cases} \bigotimes_{i=1}^{n} Sat(G_i), & \text{se } (G_1 \ldots G_i \ldots G_n) \mapsto^{and} G \\ \bigoplus_{i=1}^{n} Sat(G_1), & \text{se } (G_1 \ldots G_i \ldots G_n) \mapsto^{or} G \\ max(Sat(G_i) \cdot w), & \text{per ogni relazione } R_i : G_i \mapsto^{+S} G \\ max(Sat(G_i)), & \text{per ogni relazione } R : G_i \mapsto^{++S} G \\ max(Den(G_i)), & \text{per ogni relazione } R : G_i \mapsto^{--D} G \\ max(Den(G_i) \cdot w), & \text{per ogni relazione } R : G_i \mapsto^{-D} G \end{cases}$$

$$Den(G) = max \begin{cases} \bigoplus_{i=1}^{n} Den(G_1), & \text{se } (G_1 \ldots G_i \ldots G_n) \mapsto^{and} G \\ \bigotimes_{i=1}^{n} Den(G_1), & \text{se } (G_1 \ldots G_i \ldots G_n) \mapsto^{or} G \\ max(Den(G_i) \cdot w), & \text{per ogni relazione } R_i : G_i \mapsto^{+D} G \\ max(Den(G_i)), & \text{per ogni relazione } R : G_i \mapsto^{++D} G \\ max(Sat(G_i)), & \text{per ogni relazione } R : G_i \mapsto^{--S} G \\ max(Sat(G_i) \cdot w), & \text{per ogni relazione } R : G_i \mapsto^{-S} G \end{cases}$$

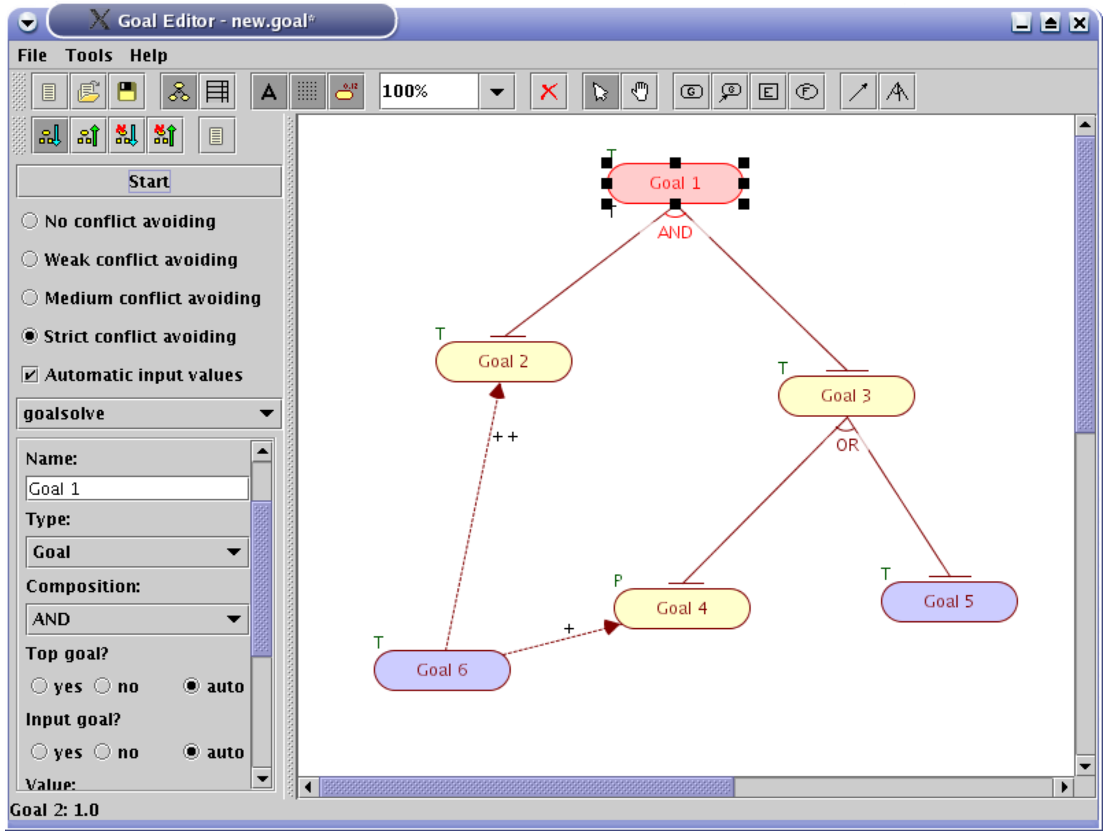
G(input):

$$Sat(G) \geq max \begin{cases} \bigotimes_{i=1}^{n} Sat(G_i), & \text{se } (G_1 \ldots G_i \ldots G_n) \mapsto^{and} G \\ \bigoplus_{i=1}^{n} Sat(G_1), & \text{se } (G_1 \ldots G_i \ldots G_n) \mapsto^{or} G \\ max(Sat(G_i) \cdot w), & \text{per ogni relazione } R_i : G_i \mapsto^{+S} G \\ max(Sat(G_i)), & \text{per ogni relazione } R : G_i \mapsto^{++S} G \\ max(Den(G_i)), & \text{per ogni relazione } R : G_i \mapsto^{--D} G \\ max(Den(G_i) \cdot w), & \text{per ogni relazione } R : G_i \mapsto^{-D} G \end{cases}$$

$$Den(G) \geq max \begin{cases} \bigoplus_{i=1}^{n} Den(G_1), & \text{se } (G_1 \ldots G_i \ldots G_n) \mapsto^{and} G \\ \bigotimes_{i=1}^{n} Den(G_1), & \text{se } (G_1 \ldots G_i \ldots G_n) \mapsto^{or} G \\ max(Den(G_i) \cdot w), & \text{per ogni relazione } R_i : G_i \mapsto^{+D} G \\ max(Den(G_i)), & \text{per ogni relazione } R : G_i \mapsto^{++D} G \\ max(Sat(G_i)), & \text{per ogni relazione } R : G_i \mapsto^{--S} G \\ max(Sat(G_i) \cdot w), & \text{per ogni relazione } R : G_i \mapsto^{-S} G \end{cases}$$

Goal Reasoning Tool

- http://troposproject.org/tools/grtool/



Constraint Goal Reasoning Tool