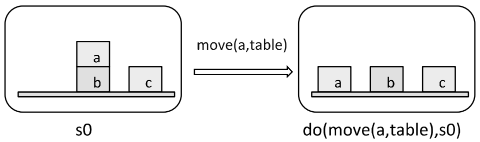
Situation Calculus



$$\underbrace{\forall x \forall y \forall s \, (\text{holds(clear(x),s)} \land (\text{holds(clear(y),s)} \lor \text{y=table})}_{\text{Προϋποθέσεις}} \Rightarrow \underbrace{\text{holds(on(x,y), do(move(x,y),s))}}_{\text{Αποτέλεσμα}})$$

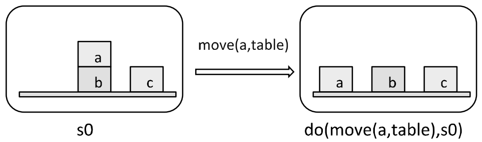
⇨ Effect Axiom

$$\forall x \forall y \forall z \forall w \forall s \, (\text{holds(on(x,y), s)} \lor x \neq z) \Rightarrow \text{holds(on(x,y), do(move(z,w),s))})$$

$$\forall x \forall y \forall z \forall w \, (\text{holds(color(x,y), s)} \Rightarrow \text{holds(color(x,y), do(move(z,w),s))})$$

⇨ Frame Axioms

Situation Calculus



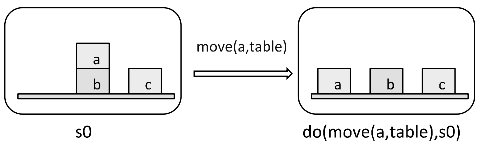
holds(on(a,b), s0)
holds(on(b, table), s0)
holds(on(c, table), s0)

Το σύμπαν μιας ερμηνείας για προτάσεις situation calculus συμπεριλαμβάνει, εκτός από στοιχεία όπως a, b, c, κλπ, και ***καταστάσεις*** (πχ σ0, σ1, σ2, . . .) και ενέργειες (πχ. ε1, ε2, . . .).

Το on(x,y) είναι ***συνάρτηση*** που επιστρέφει ένα σύνολο καταστάσεων (διαισθητικά, τις καταστάσεις στις οποίος ο κύβος x βρίσκεται πάνω στο y). Τέτοιες συναρτήσεις ονομάζονται ***fluents***.

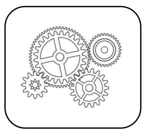
Το **holds(f,s)** είναι κατηγόρημα που δηλώνει πως η κατάσταση s ανήκει στο αποτέλεσμα του fluent f (διαισθητικά, το f ισχύει στην s.

Συμπερασμός με Situation Calculus



__Περιγραφή Μικρόκοσμου__:
Αρχική Κατάσταση
Περιορισμοί Ακεραιότητας
Effect Axioms
Frame Axioms

⇨

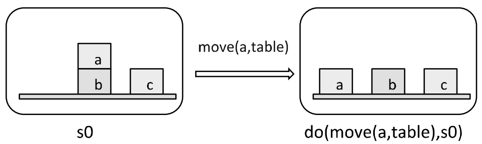


⇨ Απάντηση
(πχ «do(move(c,a), do(mova(a,table), s0))»)

⇧ Query
(πχ «πως θα φτιάξω πράσινο πύργο με βάση το α;»)

Frame Problem: Είναι (πρακτικά) πολύ δύσκολο να απαριθμήσει κανείς για κάθε ενέργεια όλα όσα **δεν αλλάζουν**.

Situation Calculus



holds(on(a,b), s0)
holds(on(b, table), s0)
holds(on(c, table), s0)
holds(color(a,green),s0)
holds(color(b,yellow),s0)
holds(color(c,green),s0)

⇨ Αρχική κατάσταση

$\forall x \, (\text{block}(x) \Leftrightarrow x=a \lor x=b \lor x=c)$
$\forall x \forall y \forall s \, (\text{holds(on(x,y), s)} \Rightarrow (\text{block}(x) \land (y=\text{table} \lor \text{block}(y))))$
$a \neq b \neq c \neq s0$
$\forall x \forall s \, (\text{holds(clear(x), s)} \Leftrightarrow \neg \exists y \, (\text{holds(on(y,x), s)}))$
$\forall x \forall y \forall s \, (\text{holds(on(x,y), s)} \land \text{holds(on(x,z), s)}) \Rightarrow y=z)$
$\forall x \forall y \forall s \, (\text{holds(on(x,y), s)} \land \text{holds(on(z,y), s)} \land y \neq \text{table} \Rightarrow x=z)$
$\forall x \forall s (\text{block}(x) \Rightarrow \exists y (\text{holds(on(x,y), s)})$

⇨ Περιορισμοί Ακεραιότητας

Συμπερασμός – Μέθοδος Green

Συζευκτική Κανονική Μορφή

1. { ¬innocent(a), friend(b,d) }
2. { ¬innocent(a), hates(c,d) }
3. { ¬innocent(b), ¬in_town(b) }
4. { ¬innocent(b), ¬knows(b,d) }
5. { ¬innocent(c), with(a,d) }
6. { ¬innocent(c), with(b,d) }
7. { ¬with(x,d), in_town(x) }
8. { ¬friend(x,y), knows(x,y) }
9. { ¬hates(x,y), knows(x,y) }
10. { ¬innocent(a), ¬innocent(b), ¬innocent(c) }
11. { innocent(a), innocent(b) }
12. { innocent(a), innocent(c) }
13. { innocent(b), innocent(c) }

Άρνηση Αποδεικτέου και Προσθήκη Απάντησης:

14. { innocent(x), Ans(x) }

Αναγωγή

15. {¬in_town(b), Ans(b) }	(14), (3)
16. { ¬with(b,d), Ans(b) }	(15), (7), [x|b]
17. { ¬innocent(c), Ans(b) }	(16), (6)
18. { innocent(a), Ans(b) }	(17), (12)
19. { ¬knows(b,d), Ans(b) }	(14), (4), [x|b]
20. { ¬friend(b,d), Ans(b) }	(19), (8), [x|b, y|d]
21. { ¬innocent(a), Ans(b) }	(20), (1)
22. { innocent(c), Ans(b) }	(21), (12)
23. { Ans(b) }	(17), (22)

Συμπερασμός – Μέθοδος Green

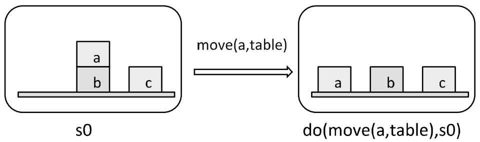
Στην πόλη έγινε ένας φόνος. Το θύμα ήταν ο David. Η αστυνομία συνέλαβε τρεις υπόπτους, τον Abbott, τον Babbitt και τον Cabot. Επίσης, υποθέτει ότι οι δύο που είναι αθώοι λένε την αλήθεια.

Οι ύποπτοι κατέθεσαν τα ακόλουθα:

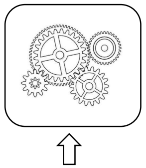
- Ο Abbott ισχυρίζεται ότι ο Babbitt ήταν φίλος του David και ότι ο Cabot μισούσε τον David.
  innocent(a) ⇒ friend(b,d)
  innocent(a) ⇒ hates(c,d)
- Ο Babbitt αρνήθηκε ότι ήταν στην πόλη την ημέρα του φόνου και επίσης είπε ότι δεν γνώριζε τον David.
  innocent(b) ⇒ ¬in_town(b)
  innocent(b) ⇒ ¬knows(b,d)
- Ο Cabot κατέθεσε ότι είδε τον Abbott και τον Babbitt να είναι με τον David λίγο πριν το έγκλημα.
  innocent(c) ⇒ with(a,d)
  innocent(c) ⇒ with(b,d)
- Η αστυνομία είναι βέβαιη ότι ακριβώς ένας από τους Abbott, Babbitt και Cabot είναι ο ένοχος.
  ¬innocent(a) ∨ ¬innocent(b) ∨ ¬innocent(c)
  (innocent(a) ∧ innocent(b)) ∨ (innocent(a) ∧ innocent(c)) ∨ (innocent(b) ∧ innocent(c))

Ποιος είναι ο δολοφόνος;

Situation Calculus και Μέθοδος Green



Περιγραφή Μικρόκοσμου:
Αρχική Κατάσταση
Περιορισμοί Ακεραιότητας
Effect Axioms
~~Frame Axioms~~



Ans(do(move(c,a), do(mova(a,table), s0)))

«πως θα φτιάξω πράσινο πύργο με βάση το α;»

∀s(goal(s) ⇔ holds(on(c,a),s) ∧ holds(on(a,table),s) ∧ holds(on(b,table),s))

{ ¬goal(s), Ans(s) }

Frame Problem: Είναι (πρακτικά) πολύ δύσκολο να απαριθμήσει κανείς για κάθε ενέργεια όλα όσα **δεν αλλάζουν**.

Συμπερασμός – Μέθοδος Green

innocent(a) ⇒ friend(b,d)
innocent(a) ⇒ hates(c,d).
innocent(b) ⇒ ¬in_town(b)
innocent(b) ⇒ ¬knows(b,d)
innocent(c) ⇒ with(a,d)
innocent(c) ⇒ with(b,d)
¬innocent(a) ∨ ¬innocent(b) ∨ ¬innocent(c)
(innocent(a) ∧ innocent(b)) ∨ (innocent(a) ∧ innocent(c)) ∨ (innocent(b) ∧ innocent(c))
∀x (with(x,d) ⇒ in_town(x))
∀x∀y (friend(x,y) ⇒ knows(x,y))
∀x∀y (hates(x,y) ⇒ knows(x,y))

Σύνολα Απαντήσεων (Answer Sets)

Η **μείωση** (reduct) ενός λογικού προγράμματος Π ως προς μια ερμηνεία Μ, που συμβολίζεται Π^M ορίζεται ως εξής:

$$\Pi^M = \{\ head(r) :\text{-}\ body(r)^+ \mid r \in \Pi \text{ και } body(r)^- \cap M = \emptyset\ \}$$

Παράδειγμα:

$$\Pi = \left\{ \begin{array}{l} a. \\ c :\text{-}\ not\ b,\ not\ d. \\ d :\text{-}\ a,\ not\ c. \end{array} \right\}$$

~~M = { a }~~ $\Pi^M = \left\{ \begin{array}{l} a. \\ c. \\ d :\text{-}\ a. \end{array} \right\}$

✔ M = { a, c } $\Pi^M = \left\{ \begin{array}{l} a. \\ c. \end{array} \right\}$

~~M = { a, b, c, d }~~ $\Pi^M = \left\{ \begin{array}{l} a. \end{array} \right\}$

Μια ερμηνεία Μ είναι **answer set** ενός λογικού προγράμματος Π ανν η Μ είναι stable model του Π^M.

Ο Κόσμος των Κύβων σε ASP

```
% Αντικείμενα Μικρόκοσμου
block(a;b;c).
object(table).
object(X) :- block(X).

% Αρχική Κατάσταση
on(a,b,0). on(b,table,0). on(c,table,0).
color(a,green,0). color(b,yellow,0). color(c,green,0).

% Στοχος
#const n=2.
goal :- on(c,a,n), on(a,table,n), on(b,table,n).
:- not goal.

% Προϋποθέσεις μετακίνησης
:- on(X,Y,T), move(Y,_,T), T<n.
:- block(Y), on(_,Y,T), move(_,Y,T), T<n.

% Effect Axiom
on(X,Y,T+1) :- move(X,Y,T), T<n.
```

```
% Frame Axiom
on(X,Y,T+1) :- on(X,Y,T), not move(X,_,T), T<n.
color(X,Y,T+1) :- color(X,Y,T), T<n.

% Επιλογή Κινήσεων
1 {move(X,Y,T): block(X), object(Y)} 1 :- T = 0..n-1.

#show move/3.
```

Παραδείγματα

$\Pi = \left\{ \begin{array}{l} a :\text{-}\ a. \\ b :\text{-}\ not\ a. \end{array} \right\}$ M = { b } ✔ $\Pi^M = \left\{ \begin{array}{l} a :\text{-}\ a. \\ b. \end{array} \right\}$

$\Pi = \left\{ \begin{array}{l} a :\text{-}\ not\ b. \\ b :\text{-}\ not\ a. \end{array} \right\}$ M = { a } ✔ $\Pi^M = \left\{ \begin{array}{l} a. \end{array} \right\}$ M = { b } ✔ $\Pi^M = \left\{ \begin{array}{l} b. \end{array} \right\}$

$\Pi = \left\{ a :\text{-}\ not\ a. \right\}$ M = { } **X** $\Pi^M = \left\{ a. \right\}$ M = { a } **X** $\Pi^M = \left\{ \ \right\}$

Θετικό Λογικό Πρόγραμμα

- Ένα λογικό πρόγραμμα ονομάζεται ***θετικό*** δεν περιέχει κανόνες που να έχουν στο σώμα τους αρνήσεις (για όλους τους κανόνες r του προγράμματος, $body(r)^- = \emptyset$).
- Στα πλαίσια του λογικού προγραμματισμού θα ονομάζουμε ***ερμηνεία*** Μ ένα σύνολο από ***ground atoms***, π.χ. M = { }, M = { a, c }, M = { move(a,table,0), move(c,a,1), color(a,green) }. Τα ground atoms που περιέχονται σε μια ερμηνεία Μ είναι αυτά που επαληθεύονται από την ερμηνεία – όλα τα υπόλοιπα ground atoms εκλαμβάνονται ως ψευδή.
- Μια ερμηνεία Μ είναι ***μοντέλο*** για ένα θετικό λογικό πρόγραμμα Π όταν ικανοποιεί όλους τους κανόνες του: για κάθε κανόνα r του προγράμματος, εφόσον $body(r)^+ \subseteq M$, τότε $head(r) \in M$.
- Το ***ελάχιστο*** μοντέλο ενός θετικού λογικού προγράμματος Π ονομάζεται ***σταθερό μοντέλο*** του Π.

Παράδειγμα:

Πρόγραμμα	Ερμηνείες	
a :- b.	{ a, b }	(μοντέλο)
b :- a	{ a }	**X**
	{ b }	**X**
	{ }	(σταθερό μοντέλο)