Τεχνητή Νοημοσύνη ΙΙ

Παύλος Πέππας

Τμήμα Ηλεκτρολόγων Μηχανικών

και Τεχνολογίας Υπολογιστών

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.

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)

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.

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)

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.

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)

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.

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))

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.
  $innocent(a) \Rightarrow friend(b,d)$
  $innocent(a) \Rightarrow hates(c,d)$
- Ο Babbitt αρνήθηκε ότι ήταν στην πόλη την ημέρα του φόνου και επίσης είπε ότι δεν γνώριζε τον David.
  $innocent(b) \Rightarrow \neg in\_town(b)$
  $innocent(b) \Rightarrow \neg knows(b,d)$
- Ο Cabot κατέθεσε ότι είδε τον Abbott και τον Babbitt να είναι με τον David λίγο πριν το έγκλημα.
  $innocent(c) \Rightarrow with(a,d)$
  $innocent(c) \Rightarrow with(b,d)$
- Η αστυνομία είναι βέβαιη ότι ακριβώς ένας από τους Abbott, Babbitt και Cabot είναι ο ένοχος.
  $\neg innocent(a) \vee \neg innocent(b) \vee \neg innocent(c)$
  $(innocent(a) \wedge innocent(b)) \vee (innocent(a) \wedge innocent(c)) \vee (innocent(b) \wedge innocent(c))$

Ποιος είναι ο δολοφόνος;

Συμπερασμός – Μέθοδος Green

innocent(a) $\Rightarrow$ friend(b,d)

innocent(a) $\Rightarrow$ hates(c,d).

innocent(b) $\Rightarrow$ ¬in_town(b)

innocent(b) $\Rightarrow$ ¬knows(b,d)

innocent(c) $\Rightarrow$ with(a,d)

innocent(c) $\Rightarrow$ with(b,d)

¬innocent(a) $\vee$ ¬innocent(b) $\vee$ ¬innocent(c)

(innocent(a) $\wedge$ innocent(b)) $\vee$ (innocent(a) $\wedge$ innocent(c)) $\vee$ (innocent(b) $\wedge$ innocent(c))

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))

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) }

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) }

Συμπερασμός – Μέθοδος Green

<u>Συζευκτική Κανονική Μορφή</u>

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) }

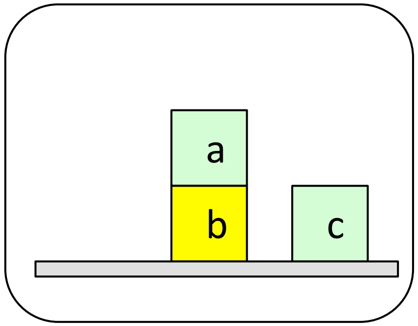
<u>Άρνηση Αποδεικτέου και Προσθήκη Απάντησης</u>:

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

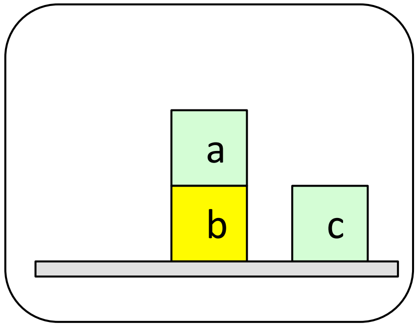
<u>Αναγωγή</u>

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 της Δράσης
(*Reasoning about Action*)

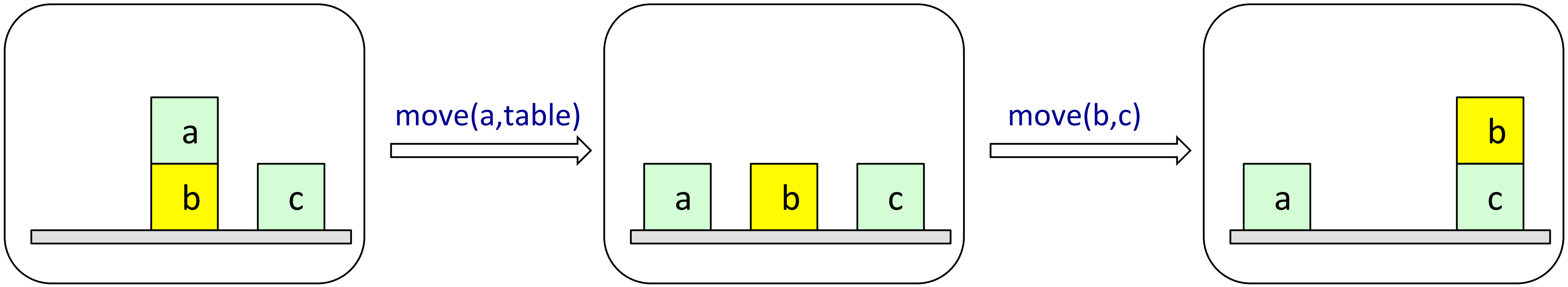


Συμπερασμός της Δράσης
(*Reasoning about Action*)



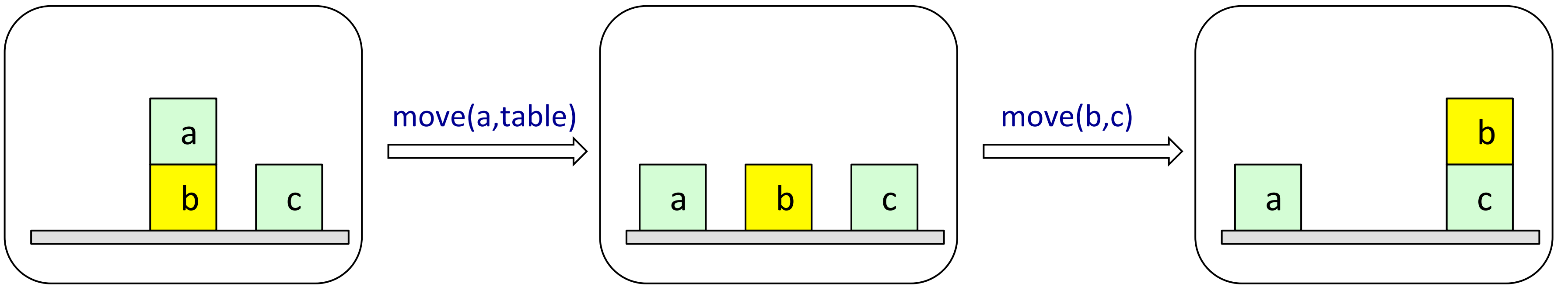
Στόχος: Το b πάνω στο c.

Συμπερασμός της Δράσης
(*Reasoning about Action*)

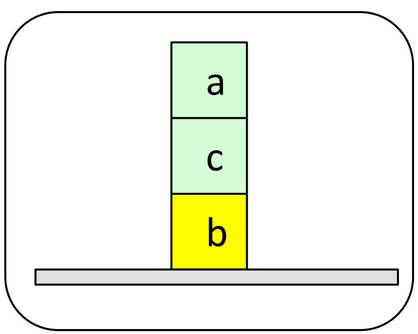


Στόχος: Το b πάνω στο c.

Συμπερασμός της Δράσης
(*Reasoning about Action*)

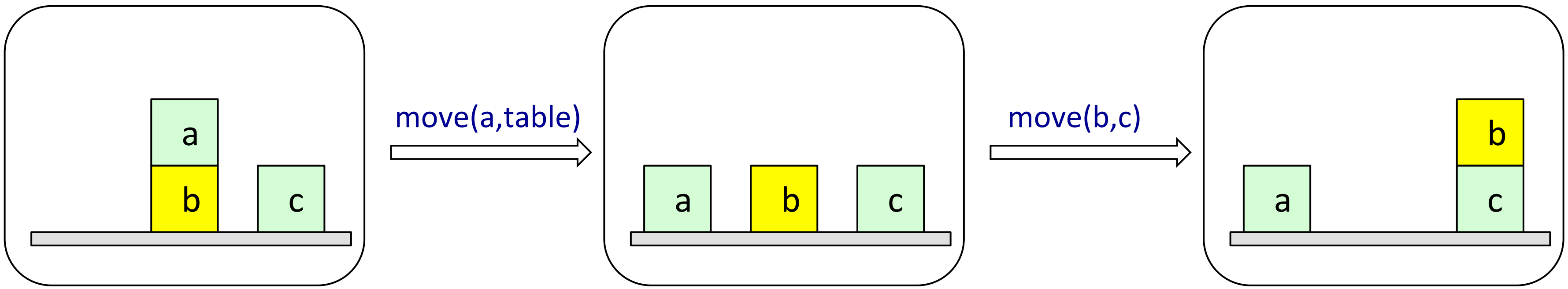


Στόχος: Το b πάνω στο c.

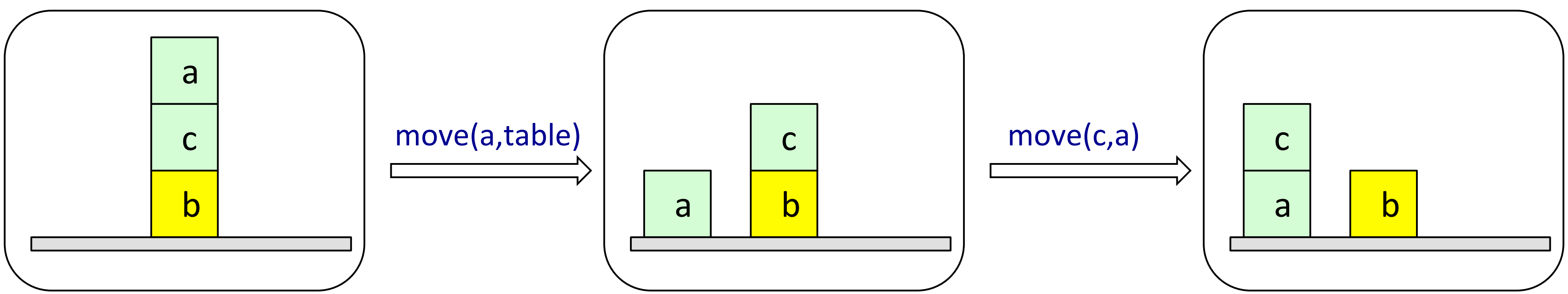


Στόχος: Πύργος μόνο με πράσινους κύβους.

Συμπερασμός της Δράσης
(*Reasoning about Action*)

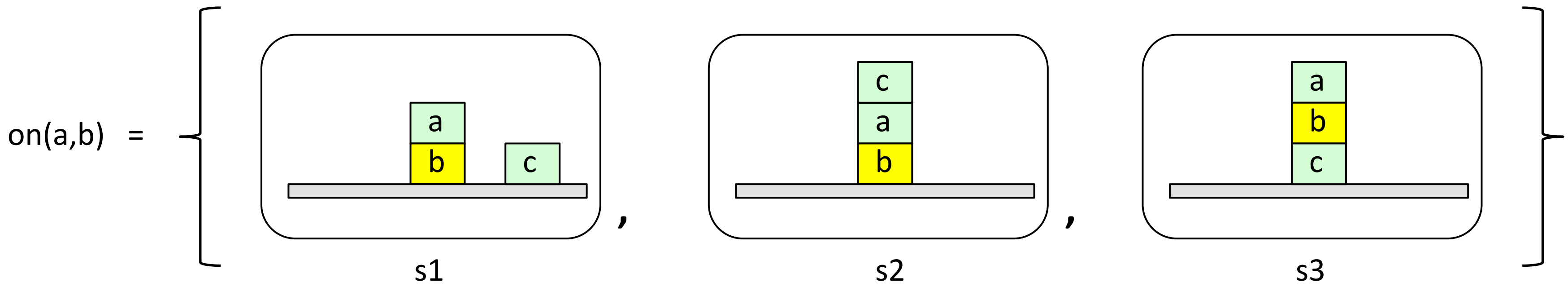


Στόχος: Το b πάνω στο c.



Στόχος: Πύργος μόνο με πράσινους κύβους.

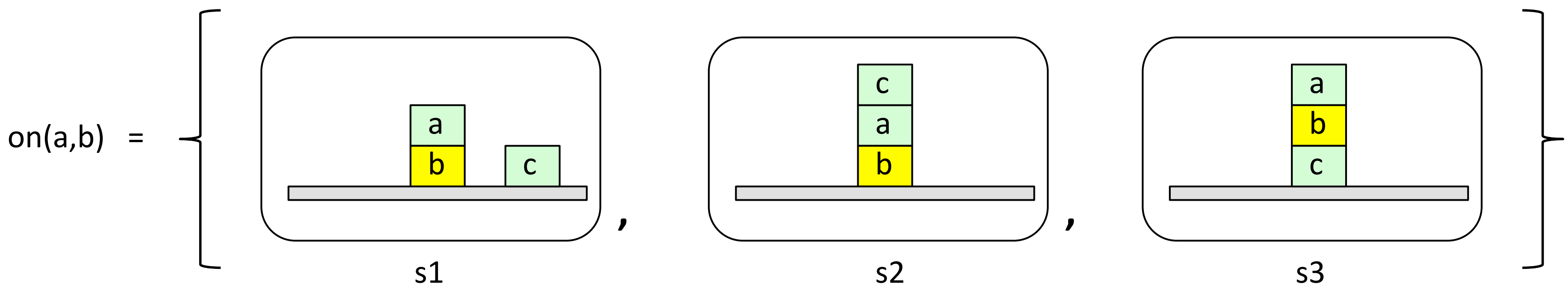
Situation Calculus



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, . . .).

Συναρτήσεις όπως το on(x,y) ονομάζονται ***fluents*** και έχουν ως τιμή ένα σύνολο καταστάσεων. Π.χ. η τιμή του on(x,y) είναι το σύνολο καταστάσεων όπου ο κύβος x βρίσκεται πάνω στο y.

Situation Calculus



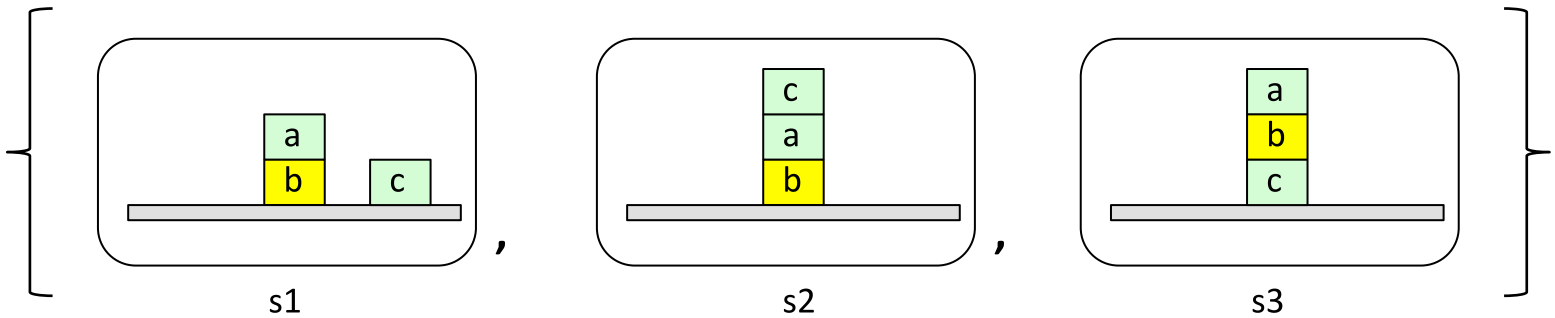
Το σύμπαν μιας ερμηνείας για προτάσεις situation calculus συμπεριλαμβάνει, εκτός από στοιχεία όπως a, b, c, κλπ, και ***καταστάσεις*** (πχ σ0, σ1, σ2, . . .) και ενέργειες (πχ. ε1, ε2, . . .).

Συναρτήσεις όπως το on(x,y) ονομάζονται ***fluents*** και έχουν ως τιμή ένα σύνολο καταστάσεων. Π.χ. η τιμή του on(x,y) είναι το σύνολο καταστάσεων όπου ο κύβος x βρίσκεται πάνω στο y.

Το **holds(*f*,s)** είναι κατηγόρημα που δηλώνει πως η κατάσταση s ανήκει στο σύνολο του fluent *f* (διαισθητικά, το *f* ισχύει στην s). Π.χ. holds(on(a,b), s1).

Situation Calculus

on(a,b) =



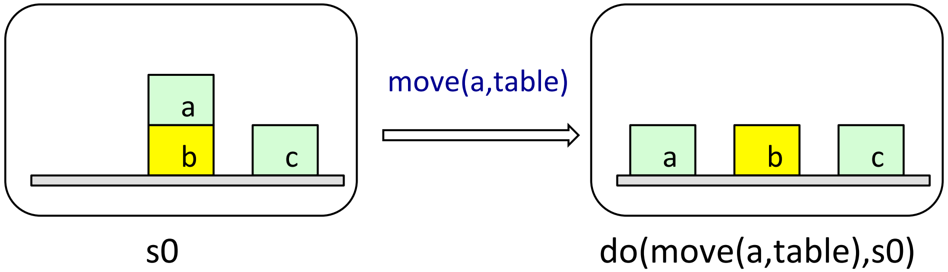
Το σύμπαν μιας ερμηνείας για προτάσεις situation calculus συμπεριλαμβάνει, εκτός από στοιχεία όπως a, b, c, κλπ, και ***καταστάσεις*** (πχ σ0, σ1, σ2, . . .) και ενέργειες (πχ. ε1, ε2, . . .).

Συναρτήσεις όπως το on(x,y) ονομάζονται ***fluents*** και έχουν ως τιμή ένα σύνολο καταστάσεων. Π.χ. η τιμή του on(x,y) είναι το σύνολο καταστάσεων όπου ο κύβος x βρίσκεται πάνω στο y.

Το **holds(*f*,s)** είναι κατηγόρημα που δηλώνει πως η κατάσταση s ανήκει στο σύνολο του fluent *f* (διαισθητικά, το *f* ισχύει στην s). Π.χ. holds(on(a,b), s1).

Το **do(ε,s)** είναι συνάρτηση με τιμή την κατάσταση που προκύπτει από την εφαρμογή της ενέργειας ε στην κατάσταση s. Π.χ. do(move(c,a), s1) = s2.

Situation Calculus



holds(on(a,b), s0)

holds(on(b, table), s0)

holds(on(c, table), s0)

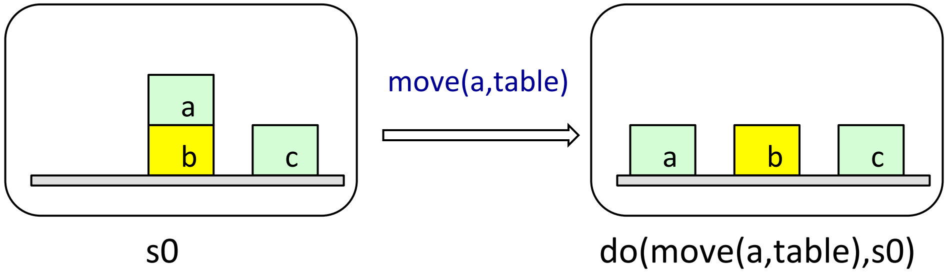
holds(color(a,green),s0)

holds(color(b,yellow),s0)

holds(color(c,green),s0)

Αρχική κατάσταση

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)

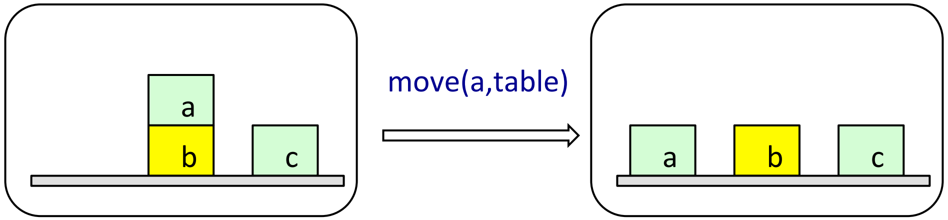
⇒ Αρχική κατάσταση

$\forall x\ (\ block(x) \Leftrightarrow x=a \lor x=b \lor x=c\ )$

$\forall x \forall y \forall s\ (\ holds(on(x,y), s) \Rightarrow (\ block(x) \land (y=table \lor block(y)\ )\ )\ )$

$a \neq b \neq c \neq s0$

Situation Calculus



s0 do(move(a,table),s0)

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

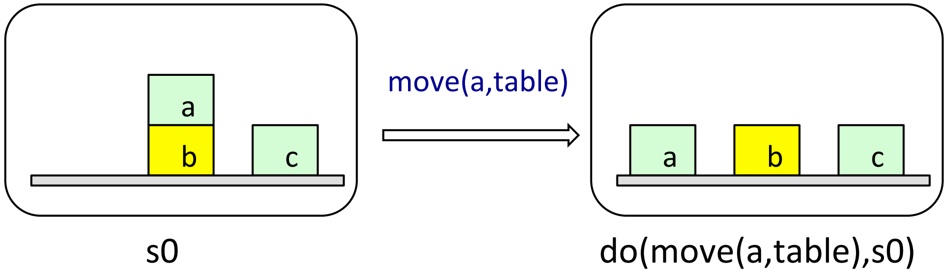
$\forall x \, (\, block(x) \Leftrightarrow x=a \lor x=b \lor x=c \,)$

$\forall x \forall y \forall s \, (\, holds(on(x,y), s) \Rightarrow (\, block(x) \land (y=table \lor block(y)\,)\,)\,)$

$a \neq b \neq c \neq s0$

$\forall x \forall s \, (\, holds(clear(x), s) \Leftrightarrow \neg \exists y \, (holds(on(y,x), s)\,)\,)$

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

$\forall x\ (\ block(x) \Leftrightarrow x=a \vee x=b \vee x=c\ )$

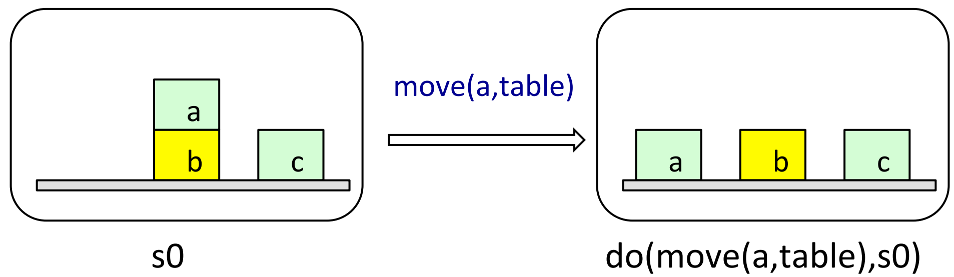
$\forall x \forall y \forall s\ (\ holds(on(x,y), s) \Rightarrow (\ block(x) \wedge (y=table \vee block(y)\ )\ )\ )$

$a \neq b \neq c \neq s0$

$\forall x \forall s\ (\ holds(clear(x), s) \Leftrightarrow \neg \exists y\ (holds(on(y,x), s)\ )\ )$

$\forall x \forall y \forall s\ (\ holds(on(x,y), s) \wedge holds(on(x,z), s)\ ) \Rightarrow y=z\ )$

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

$\forall x\ (\ block(x) \Leftrightarrow x=a \lor x=b \lor x=c\ )$

$\forall x \forall y \forall s\ (\ holds(on(x,y), s) \Rightarrow (\ block(x) \land (y=table \lor block(y)\ )\ )\ )$

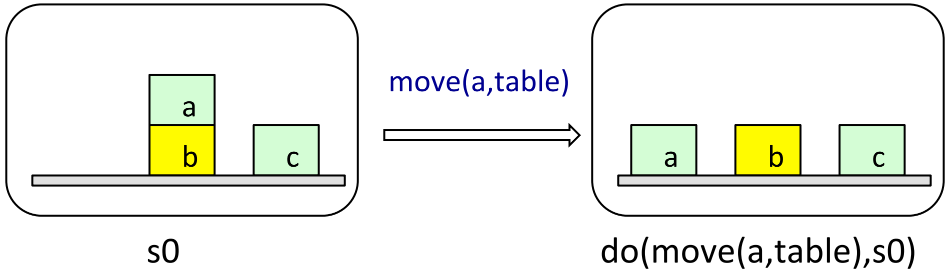
$a \neq b \neq c \neq s0$

$\forall x \forall s\ (\ holds(clear(x), s) \Leftrightarrow \neg \exists y\ (holds(on(y,x), s)\ )\ )$

$\forall x \forall y \forall s\ (\ holds(on(x,y), s) \land holds(on(x,z), s)\ ) \Rightarrow y=z\ )$

$\forall x \forall y \forall s\ (\ holds(on(x,y), s) \land holds(on(z,y), s) \land y \neq table \Rightarrow x=z\ )$

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

$\forall x\ (\ block(x) \Leftrightarrow x=a \lor x=b \lor x=c\ )$

$\forall x \forall y \forall s\ (\ holds(on(x,y), s) \Rightarrow (\ block(x) \land (y=table \lor block(y)\ )\ )\ )$

$a \neq b \neq c \neq s0$

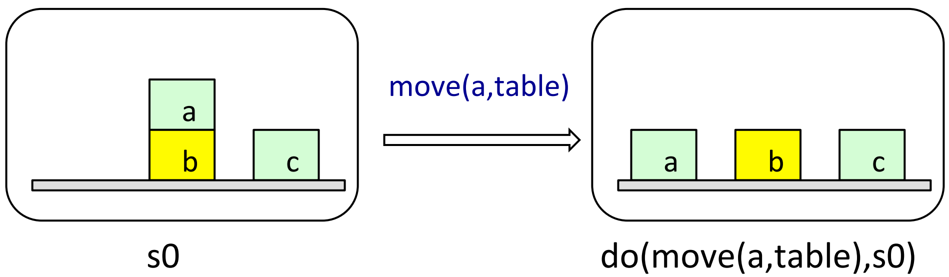
$\forall x \forall s\ (\ holds(clear(x), s) \Leftrightarrow \neg \exists y\ (holds(on(y,x), s)\ )\ )$

$\forall x \forall y \forall s\ (\ holds(on(x,y), s) \land holds(on(x,z), s)\ ) \Rightarrow y=z\ )$

$\forall x \forall y \forall s\ (\ holds(on(x,y), s) \land holds(on(z,y), s) \land y \neq table \Rightarrow x=z\ )$

$\forall x \forall s(\ block(x) \Rightarrow \exists y(\ holds(on(x,y), s)\ )\ )$

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

$\forall x \, (block(x) \Leftrightarrow x=a \lor x=b \lor x=c)$

$\forall x \forall y \forall s \, (holds(on(x,y), s) \Rightarrow (block(x) \land (y=table \lor block(y))))$

$a \neq b \neq c \neq s0$

$\forall x \forall s \, (holds(clear(x), s) \Leftrightarrow \neg \exists y \, (holds(on(y,x), s)))$

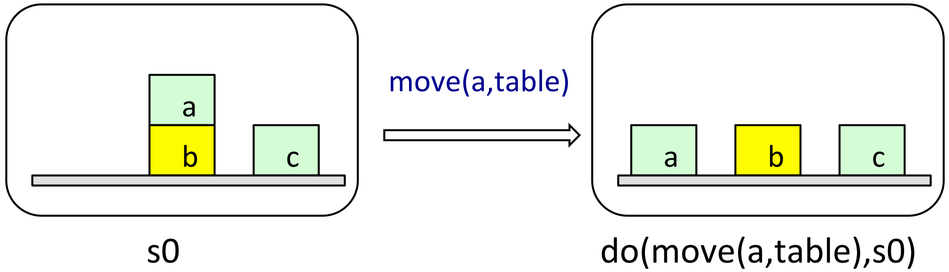
$\forall x \forall y \forall s \, (holds(on(x,y), s) \land holds(on(x,z), s)) \Rightarrow y=z)$

$\forall x \forall y \forall s \, (holds(on(x,y), s) \land holds(on(z,y), s) \land y \neq table \Rightarrow x=z)$

$\forall x \forall s(block(x) \Rightarrow \exists y(holds(on(x,y), s))$

Περιορισμοί Ακεραιότητας

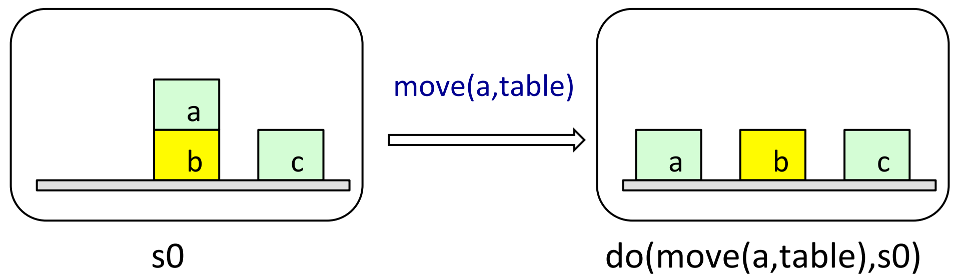
Situation Calculus



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

Effect Axiom

Situation Calculus



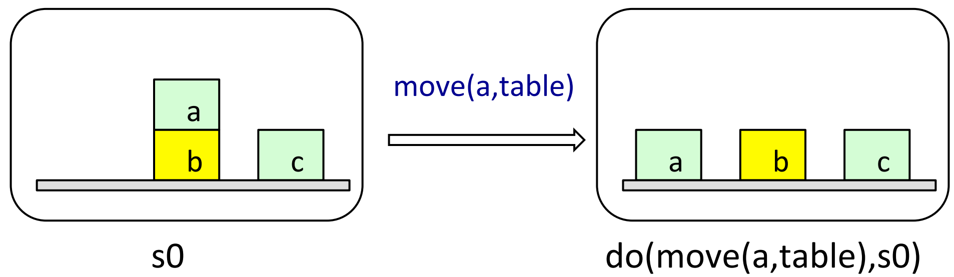
$$\forall x \forall y \forall s\ (\underbrace{holds(clear(x),s) \land (holds(clear(y),s) \lor y=table)}_{\text{Προϋποθέσεις}} \Rightarrow \underbrace{holds(on(x,y),\ do(move(x,y),s))}_{\text{Αποτέλεσμα}})$$

Effect Axiom

$$\forall x \forall y \forall z \forall w \forall s\ (holds(on(x,y),\ s) \land x \neq z) \Rightarrow holds(on(x,y),\ do(move(z,w),s))\ )$$

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

Situation Calculus



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

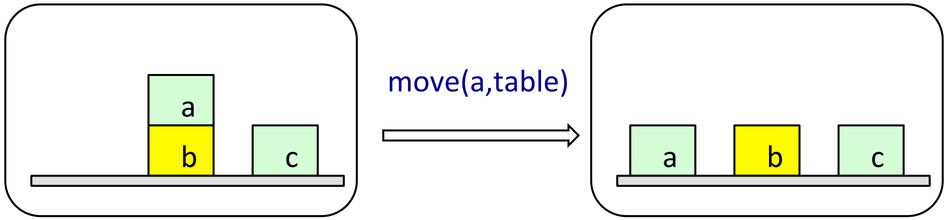
Effect Axiom

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

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

Frame Axioms

Συμπερασμός με Situation Calculus



s0

do(move(a,table),s0)

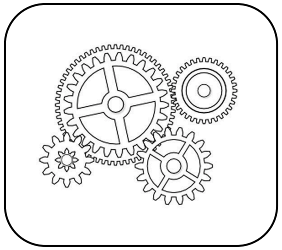
<u>Περιγραφή Μικρόκοσμου</u>:

Αρχική Κατάσταση

Περιορισμοί Ακεραιότητας

Effect Axioms

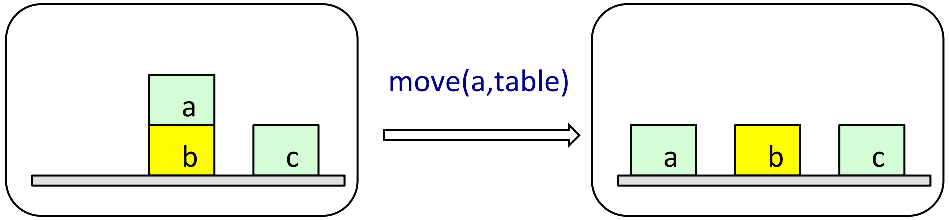
Frame Axioms



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

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

Συμπερασμός με Situation Calculus



s0 do(move(a,table),s0)

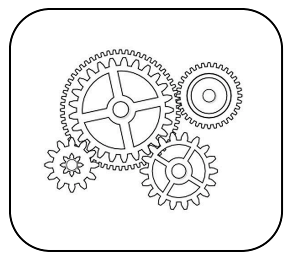
Περιγραφή Μικρόκοσμου:

Αρχική Κατάσταση

Περιορισμοί Ακεραιότητας

Effect Axioms

Frame Axioms



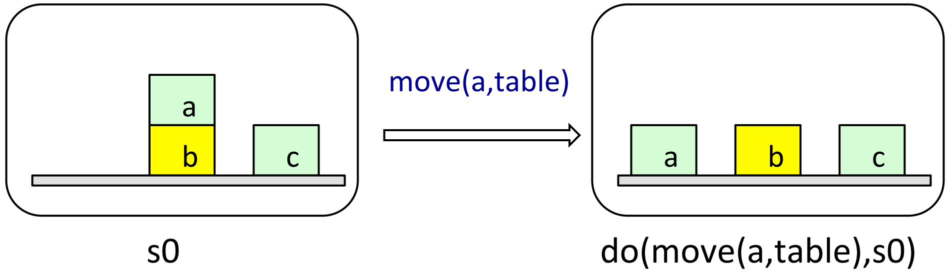
Ans(do(move(c,a), do(mova(a,table), s0)))

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

$\forall$s[goal(s) $\Leftrightarrow$ holds(on(c,a),s) $\wedge$ holds(on(a,table),s) $\wedge$ holds(on(b,table),s)]

$\exists$s[goal(s) $\wedge$ Ans(s)]

Συμπερασμός με Situation Calculus



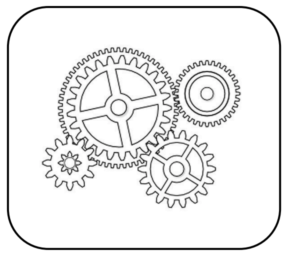
<u>Περιγραφή Μικρόκοσμου</u>:

Αρχική Κατάσταση

Περιορισμοί Ακεραιότητας

Effect Axioms

Frame Axioms



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

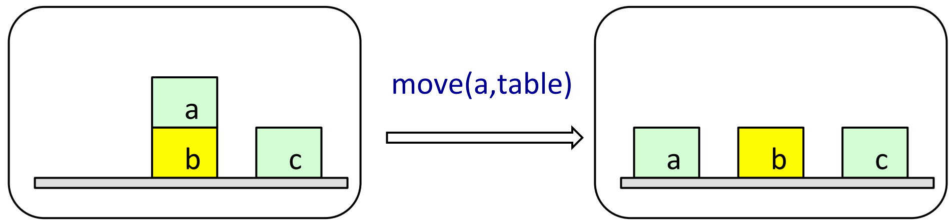
«πως θα φτιάξω πράσινο πύργο με βάση το α;»

$\forall$ s[goal(s) $\Leftrightarrow$ holds(on(c,a),s) $\wedge$ holds(on(a,table),s) $\wedge$ holds(on(b,table),s)]

$\exists$ s[goal(s) $\wedge$ Ans(s)]

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

Συμπερασμός με Situation Calculus



s0 do(move(a,table),s0)

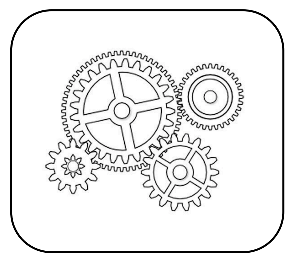
<u>Περιγραφή Μικρόκοσμου</u>:

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;»

$$\forall s[\ goal(s) \Leftrightarrow holds(on(c,a),s) \land holds(on(a,table),s) \land holds(on(b,table),s)\ ]$$

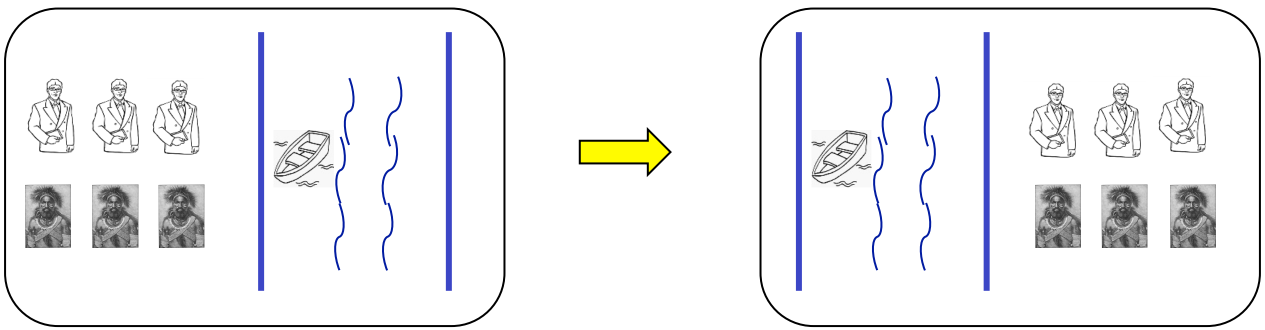
$$\exists s[\ goal(s) \land Ans(s)\ ]$$

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

Άσκηση

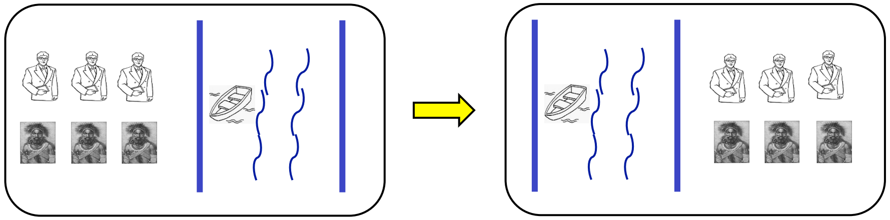
Διατυπώστε το πρόβλημα των Ιεραποστόλων-Κανιβάλων σε Situation Calculus.

Αρχική κατάσταση Τελική κατάσταση



Στη μία όχθη ενός ποταμού υπάρχουν 3 ιεραπόστολοι και 3 κανίβαλοι. Έχουν στη διάθεσή τους μία βάρκα, με την οποία μπορούν να μεταφερθούν 1 ή 2 άτομα. Το πρόβλημα είναι να μεταφερθούν και οι 6 στην άλλη όχθη, χωρίς όμως ποτέ να βρίσκονται σε κάποια όχθη περισσότεροι κανίβαλοι από ιεραποστόλους. Προφανώς η βάρκα δεν μπορεί να ταξιδέψει μόνη της.

Λύση



holds(boat(left), s0)

holds(miss(3,left),s0)

holds(cann(3,left),s0)

opposite(left,right)

opposite(right,left)

∀x∀u∀w∀s [holds(miss(x,u),s) ∧ opposite(u,w) ⇔ holds(miss(3-x,w),s)]

∀x∀u∀w∀s [holds(cann(x,u),s) ∧ opposite(u,w) ⇔ holds(cann(3-x,w),s)]

left != right != s0 != 1 != 2 != 3

Λύση

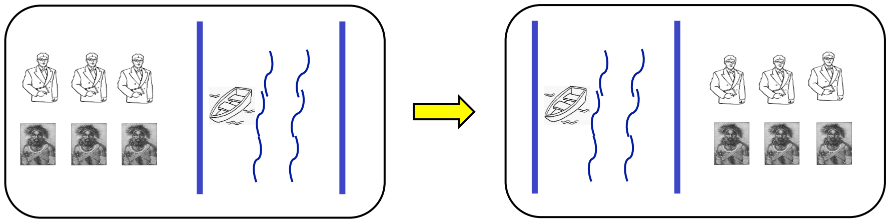
holds(boat(left), s0)

holds(miss(3,left),s0)

holds(cann(3,left),s0)

opposite(left,right)

opposite(right,left)



$\forall x \forall u \forall w \forall s$ [holds(miss(x,u),s) $\wedge$ opposite(u,w) $\Leftrightarrow$ holds(miss(3-x,w),s)]

$\forall x \forall u \forall w \forall s$ [holds(cann(x,u),s) $\wedge$ opposite(u,w) $\Leftrightarrow$ holds(cann(3-x,w),s)]

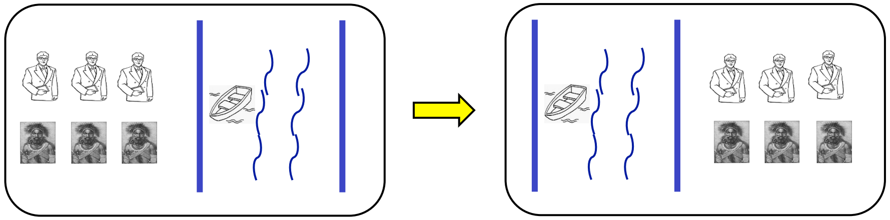
left != right != s0 != 1 != 2 != 3

Effect Axioms

$\forall x \forall y \forall u \forall w \forall s$ [holds(boat(u), s0) $\wedge$ $1 \leq x \leq 2$ $\wedge$
holds(cann(y,u),s) $\wedge$ $x \leq y$ $\wedge$ opposite(u,w) $\wedge$
holds(miss(z,u),s) $\wedge$ ($z=0$ $\vee$ $z \leq y-x$) $\Rightarrow$ holds(cann(3-y+x,w),do(moveC(x),s)) $\wedge$
holds(boat(w), do(moveC(x),s))]

$\forall x \forall y \forall u \forall w \forall s$ [holds(boat(u), s0) $\wedge$ $1 \leq x \leq 2$ $\wedge$
holds(miss(y,u),s) $\wedge$ $x \leq y$ $\wedge$ opposite(u,w) $\wedge$
holds(cann(z,u),s) $\wedge$ $z = y-x$ $\Rightarrow$ holds(miss(3-y+x,w),do(moveM(x),s)) $\wedge$
holds(boat(w), do(moveM(x),s))]

Λύση (2)



$\vdots$

$$\forall y \forall u \forall w \forall s[\ holds(boat(u), s0) \wedge holds(miss(y,u),s) \wedge holds(cann(z,u),s) \wedge$$
$$y>0 \wedge z>0 \wedge opposite(u,w) \Rightarrow holds(miss(y\text{-}1,u),do(moveCM(),s)) \wedge$$
$$holds(cann(z\text{-}1,u),do(moveCM(),s)) \wedge$$
$$holds(boat(w), do(moveCM(x),s))\ ]$$

Frame Axioms

$$\forall x \forall y \forall u \forall s[\ holds(miss(y,u),s) \Rightarrow holds(miss(y,u),do(moveC(x),s))$$

$$\forall x \forall y \forall u \forall s[\ holds(cann(y,u),s) \Rightarrow holds(cann(y,u),do(moveM(x),s))$$

Ο Κόσμος των Κύβων σε ASP

```
#const max=10.

% 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).
```

Ο Κόσμος των Κύβων σε ASP

```
#const max=10.

% 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
goal :- on(a,table,max), on(c,a,max), on(b,table,max).
:- not goal.

% Επιλογή Κινήσεων
{ move(X,Y,T): block(X), object(Y) } 1 :- T = 0..max-1.
:- not move(_,_,T), move(_,_,T+1), T = 0..max-2.
```

Ο Κόσμος των Κύβων σε ASP

```
#const max=10.

% 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
goal :- on(a,table,max), on(c,a,max), on(b,table,max).
:- not goal.

% Επιλογή Κινήσεων
{ move(X,Y,T): block(X), object(Y) } 1 :- T = 0..max-1.
:- not move(_,_,T), move(_,_,T+1), T = 0..max-2.
```

```
% Προϋποθέσεις μετακίνησης
:- on(X,Y,T), move(Y,_,T), T=0..max-1.
:- block(Y), on(_,Y,T), move(_,Y,T), T=0..max-1.

% Effect Axiom
on(X,Y,T+1) :- move(X,Y,T), T = 0..max-1.
```

Ο Κόσμος των Κύβων σε ASP

```
#const max=10.

% 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
goal :- on(a,table,max), on(c,a,max), on(b,table,max).
:- not goal.

% Επιλογή Κινήσεων
{ move(X,Y,T): block(X), object(Y) } 1 :- T = 0..max-1.
:- not move(_,_,T), move(_,_,T+1), T = 0..max-2.
```

```
% Προϋποθέσεις μετακίνησης
:- on(X,Y,T), move(Y,_,T), T=0..max-1.
:- block(Y), on(_,Y,T), move(_,Y,T), T=0..max-1.

% Effect Axiom
on(X,Y,T+1) :- move(X,Y,T), T = 0..max-1.

% Frame Axiom
on(X,Y,T+1) :- on(X,Y,T), not move(X,_,T), T = 0..max-1.
color(X,Y,T+1) :- color(X,Y,T), T = 0..max-1.

#show move/3.
```

Λογικό Πρόγραμμα

Ένα λογικό πρόγραμμα Π είναι ένα πεπερασμένο σύνολο κανόνων r της μορφής:

$$\underbrace{a_0}_{\text{Κεφαλή}} \;\; :- \;\; \underbrace{a_1, \ldots, a_m,\ \text{not } a_{m+1}, \ldots, \text{not } a_n.}_{\text{Σώμα}}$$

Λογικό Πρόγραμμα

Ένα λογικό πρόγραμμα Π είναι ένα πεπερασμένο σύνολο κανόνων r της μορφής:

$$\underbrace{a_0}_{\text{Κεφαλή}} \;:\!-\; \underbrace{a_1, \ldots, a_m,\ \text{not } a_{m+1}, \ldots, \text{not } a_n.}_{\text{Σώμα}}$$

head(r) = a_0

body(r) = $\{ a_1, \ldots, a_m,\ \text{not } a_{m+1}, \ldots, \text{not } a_n \}$

body(r)$^+$ = $\{ a_1, \ldots, a_m \}$

body(r)$^-$ = $\{ a_{m+1}, \ldots, a_n \}$

Λογικό Πρόγραμμα

Ένα λογικό πρόγραμμα Π είναι ένα πεπερασμένο σύνολο κανόνων r της μορφής:

$$\underbrace{a_0}_{\text{Κεφαλή}} \;:\!-\; \underbrace{a_1, \ldots, a_m,\ \text{not } a_{m+1}, \ldots, \text{not } a_n.}_{\text{Σώμα}}$$

$head(r) = a_0$

$body(r) = \{ a_1, \ldots, a_m,\ \text{not } a_{m+1}, \ldots, \text{not } a_n \}$

$body(r)^+ = \{ a_1, \ldots, a_m \}$

$body(r)^- = \{ a_{m+1}, \ldots, a_n \}$

Συμβολισμός:

- Το a_0. είναι συντομογραφία του a_0 :- . (κανόνας με κενό σώμα).
- Το :- $a_1, \ldots, a_m,\ \text{not } a_{m+1}, \ldots, \text{not } a_n$. δηλώνει πως ***δεν ισχύουν ταυτόχρονα*** τα $a_1, \ldots, a_m,\ \text{not } a_{m+1}, \ldots, \text{not } a_n$.

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, $body(r)^- = \emptyset$).

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.

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 του κανόνες του: για κάθε κανόνα r του προγράμματος, εφόσον $body(r)^+ \subseteq M$, τότε $head(r) \in M$.

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)$^{-}$ = ∅).

- Στα πλαίσια του λογικού προγραμματισμού θα ονομάζουμε ***ερμηνεία*** M 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 M είναι αυτά που επαληθεύονται από την ερμηνεία – όλα τα υπόλοιπα ground atoms εκλαμβάνονται ως ψευδή.

- Μια ερμηνεία M είναι ***μοντέλο*** για ένα θετικό λογικό πρόγραμμα Π όταν ικανοποιεί όλους του κανόνες του: για κάθε κανόνα r του προγράμματος, εφόσον body(r)$^{+}$ ⊆ M, τότε head(r) ∈ M.

- Το μοντέλο ενός θετικού λογικού προγράμματος Π που περιέχει ***μόνο*** όσα ground atoms παράγονται από τα facts και τους κανόνες το Π, ονομάζεται ***σταθερό μοντέλο*** του Π.

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 του κανόνες του: για κάθε κανόνα r του προγράμματος, εφόσον $body(r)^+ \subseteq M$, τότε $head(r) \in M$.

- Το μοντέλο ενός θετικού λογικού προγράμματος Π που περιέχει ***μόνο*** όσα ground atoms παράγονται από τα facts και τους κανόνες το Π, ονομάζεται ***σταθερό μοντέλο*** του Π.

Παράδειγμα:

Πρόγραμμα	Ερμηνείες
a :- b.	{ a, b }
b :- a	{ a }
	{ b }
	{ }

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*** M 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 M είναι αυτά που επαληθεύονται από την ερμηνεία – όλα τα υπόλοιπα ground atoms εκλαμβάνονται ως ψευδή.

- Μια ερμηνεία M είναι ***μοντέλο*** για ένα θετικό λογικό πρόγραμμα Π όταν ικανοποιεί όλους του κανόνες του: για κάθε κανόνα r του προγράμματος, εφόσον $body(r)^+ \subseteq M$, τότε $head(r) \in M$.

- Το μοντέλο ενός θετικού λογικού προγράμματος Π που περιέχει ***μόνο*** όσα ground atoms παράγονται από τα facts και τους κανόνες το Π, ονομάζεται ***σταθερό μοντέλο*** του Π.

Παράδειγμα:

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

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*** M 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 M είναι αυτά που επαληθεύονται από την ερμηνεία – όλα τα υπόλοιπα ground atoms εκλαμβάνονται ως ψευδή.

- Μια ερμηνεία M είναι ***μοντέλο*** για ένα θετικό λογικό πρόγραμμα Π όταν ικανοποιεί όλους του κανόνες του: για κάθε κανόνα r του προγράμματος, εφόσον $body(r)^+ \subseteq M$, τότε $head(r) \in M$.

- Το μοντέλο ενός θετικού λογικού προγράμματος Π που περιέχει ***μόνο*** όσα ground atoms παράγονται από τα facts και τους κανόνες το Π, ονομάζεται ***σταθερό μοντέλο*** του Π.

Παράδειγμα:

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

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*** M 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 M είναι αυτά που επαληθεύονται από την ερμηνεία – όλα τα υπόλοιπα ground atoms εκλαμβάνονται ως ψευδή.

- Μια ερμηνεία M είναι ***μοντέλο*** για ένα θετικό λογικό πρόγραμμα Π όταν ικανοποιεί όλους του κανόνες του: για κάθε κανόνα r του προγράμματος, εφόσον $body(r)^+ \subseteq M$, τότε $head(r) \in M$.

- Το μοντέλο ενός θετικού λογικού προγράμματος Π που περιέχει ***μόνο*** όσα ground atoms παράγονται από τα facts και τους κανόνες το Π, ονομάζεται ***σταθερό μοντέλο*** του Π.

Παράδειγμα:

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

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, $body(r)^- = \oslash$).

- Στα πλαίσια του λογικού προγραμματισμού θα ονομάζουμε ***ερμηνεία*** M 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 M είναι αυτά που επαληθεύονται από την ερμηνεία – όλα τα υπόλοιπα ground atoms εκλαμβάνονται ως ψευδή.

- Μια ερμηνεία M είναι ***μοντέλο*** για ένα θετικό λογικό πρόγραμμα Π όταν ικανοποιεί όλους του κανόνες του: για κάθε κανόνα r του προγράμματος, εφόσον $body(r)^+ \subseteq M$, τότε $head(r) \in M$.

- Το μοντέλο ενός θετικού λογικού προγράμματος Π που περιέχει ***μόνο*** όσα ground atoms παράγονται από τα facts και τους κανόνες το Π, ονομάζεται ***σταθερό μοντέλο*** του Π.

Παράδειγμα:

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

Σύνολα Απαντήσεων (Answer Sets)

Η **μείωση** (reduct) ενός λογικού προγράμματος Π ως προς μια ερμηνεία Μ, που συμβολίζεται $\mathbf{\Pi^M}$ ορίζεται ως εξής:

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

Παράδειγμα:

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

Σύνολα Απαντήσεων (Answer Sets)

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

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

Παράδειγμα:

M = { a }

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

Σύνολα Απαντήσεων (Answer Sets)

Η **μείωση** (reduct) ενός λογικού προγράμματος Π ως προς μια ερμηνεία Μ, που συμβολίζεται $\mathbf{\Pi^M}$ ορίζεται ως εξής:

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

Παράδειγμα:

$M = \{ a \}$ $\qquad$ $M = \{ a, c \}$

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

Σύνολα Απαντήσεων (Answer Sets)

Η **μείωση** (reduct) ενός λογικού προγράμματος Π ως προς μια ερμηνεία M, που συμβολίζεται $\mathbf{\Pi^M}$ ορίζεται ως εξής:

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

Παράδειγμα:

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

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 }

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

Σύνολα Απαντήσεων (Answer Sets)

Η **μείωση** (reduct) ενός λογικού προγράμματος Π ως προς μια ερμηνεία Μ, που συμβολίζεται $\mathbf{\Pi^M}$ ορίζεται ως εξής:

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

Παράδειγμα:

$M = \{ a \}$ $M = \{ a, c \}$ $M = \{ a, b, c, d \}$

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

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

Σύνολα Απαντήσεων (Answer Sets)

Η **μείωση** (reduct) ενός λογικού προγράμματος Π ως προς μια ερμηνεία M, που συμβολίζεται $\mathbf{\Pi^M}$ ορίζεται ως εξής:

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

Παράδειγμα:

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

$\cancel{M = \{ a \}}$

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

$M = \{ a, c \}$

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

$M = \{ a, b, c, d \}$

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

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

Σύνολα Απαντήσεων (Answer Sets)

Η **μείωση** (reduct) ενός λογικού προγράμματος Π ως προς μια ερμηνεία Μ, που συμβολίζεται $\mathbf{\Pi^{M}}$ ορίζεται ως εξής:

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

Παράδειγμα:

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

~~M = { a }~~

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

✔

M = { a, c }

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

M = { a, b, c, d }

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

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

Σύνολα Απαντήσεων (Answer Sets)

Η **μείωση** (reduct) ενός λογικού προγράμματος Π ως προς μια ερμηνεία Μ, που συμβολίζεται $\mathbf{\Pi^M}$ ορίζεται ως εξής:

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

Παράδειγμα:

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

~~M = { a }~~

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

✔

M = { a, c }

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

~~M = { a, b, c, d }~~

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

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

Παραδείγματα

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

Παραδείγματα

$M = \{ b \}$ ✔

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

Παραδείγματα

$M = \{ b \}$ ✔

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

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

Παραδείγματα

$M = \{ b \}$ ✔

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

$M = \{ a \}$ ✔ $\qquad$ $M = \{ b \}$ ✔

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

Παραδείγματα

$M = \{\ b\ \}$ ✔

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

$M = \{\ a\ \}$ ✔ $\qquad M = \{\ b\ \}$ ✔

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

$$\Pi = \left\{ \text{a :- not a.} \right\}$$

Παραδείγματα

$M = \{ b \}$ ✔

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

$M = \{ a \}$ ✔ $\quad$ $M = \{ b \}$ ✔

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

$M = \{ \}$ X $\quad$ $M = \{ a \}$ X

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