Ο Κανόνας Συμπερασμού της Ανάλυσης

- Ο κανόνας συμπερασμού της ανάλυσης στην προτασιακή λογική και τη λογική πρώτης τάξης.
- Χρήσεις του κανόνα συμπερασμού της ανάλυσης σε αποδείξεις μη-ικανοποιησιμότητας, λογικής κάλυψης και εγκυρότητας.
- Ανάλυση με ισότητα
- Συστήματα που χρησιμοποιούν ανάλυση (Prolog, συστήματα λογικού προγραμματισμού κλπ.)
- Η ιστορία της Λογικής

Αποδείξεις με Ανάλυση

Ο κανόνας συμπερασμού της **ανάλυσης (resolution)** για την **προτασιακή λογική** είναι ο εξής:

$$\frac{\alpha \vee \beta, \;\; \neg\beta \vee \gamma}{\alpha \vee \gamma} \qquad \text{ή ισοδύναμα} \qquad \frac{\neg\alpha \Rightarrow \beta, \;\; \beta \Rightarrow \gamma}{\neg\alpha \Rightarrow \gamma}$$

Στις διαλέξεις για την προτασιακή λογική καλύψαμε αναλυτικά την περίπτωση αυτή.

Ο Κανόνας Συμπερασμού της Ανάλυσης

Ο κανόνας συμπερασμού της ανάλυσης για την **λογική πρώτης τάξης** είναι ο εξής:

$$\frac{p_1 \vee \cdots \vee p_j \cdots \vee p_m, \quad q_1 \vee \cdots \vee q_k \cdots \vee q_n}{SUBST(\sigma, (p_1 \vee \cdots \vee p_{j-1} \vee p_{j+1} \vee \cdots \vee p_m \vee q_1 \cdots \vee q_{k-1} \vee q_{k+1} \cdots \vee q_n))}$$

όπου $\sigma = UNIFY(p_j, \neg q_k)$.

Τα λεκτικά p_j και q_k λέγονται **συμπληρωματικά (complementary)** επειδή καθένα τους ενοποιείται με την άρνηση του άλλου.

Σαν αντικατάσταση σ συνήθως διαλέγουμε τον **πιο γενικό ενοποιητή** των p_j και $\neg q_k$.

Η διάζευξη που προκύπτει λέγεται **resolvent**.

Η ανάλυση εφαρμόζεται σε διαζεύξεις λεκτικών που έχουμε συμφωνήσει να τις ονομάζουμε **φράσεις (clauses).**

Παραδείγματα

$$\frac{\neg Rich(x) \vee Unhappy(x), \quad Rich(Me)}{Unhappy(Me)}$$

Ο MGU που χρησιμοποιήθηκε είναι $\sigma = \{x/Me\}$.

Παραδείγματα

$$\frac{P(w,y) \vee Q(y,z) \vee R(F(B),w), \quad P(x,z) \vee \neg R(F(z),C) \vee S(z)}{P(C,y) \vee Q(y,B) \vee P(x,B) \vee S(B)}$$

O MGU που χρησιμοποιήθηκε είναι $\sigma = \{z/B, w/C\}$.

Χρήση της Ανάλυσης για Αποδείξεις Λογικής Κάλυψης

Έστω ότι έχουμε μια βάση γνώσεων KB και ένα τύπο ϕ.

Πως μπορούμε να δείξουμε ότι $KB \models \phi$ χρησιμοποιώντας ανάλυση;

1. Προσθέσουμε την **άρνηση** της ϕ στην KB
2. Εφαρμόζουμε τον κανόνα της ανάλυσης όσες φορές χρειάζεται μέχρι να καταλήξουμε στην **κενή φράση (empty clause)**, δηλαδή σε μια **αντίφαση**.

   Στην περίπτωση αυτή: αφού το σύνολο φράσεων $KB \cup \{ \neg\phi \}$ είναι μη-ικανοποιήσιμο, μπορούμε να συμπεράνουμε ότι $KB \models \phi$.

Παράδειγμα

$PhD(x) \Rightarrow HighlyQualified(x)$

$\neg PhD(x) \Rightarrow EarlyEarnings(x)$

$HighlyQualified(x) \Rightarrow Rich(x)$

$EarlyEarnings(x) \Rightarrow Rich(x)$

Θα χρησιμοποιήσουμε ανάλυση για να **συμπεράνουμε** $Rich(Me)$.

Στην περίπτωσή μας λοιπόν, προσθέτουμε τον τύπο $\neg Rich(Me)$ στην KB.

Παράδειγμα

Ας γράψουμε αρχικά όλες τις προτάσεις ως **διαζεύξεις**:

$\neg PhD(x) \vee HighlyQualified(x)$

$PhD(x) \vee EarlyEarnings(x)$

$\neg HighlyQualified(x) \vee Rich(x)$

$\neg EarlyEarnings(x) \vee Rich(x)$

$\neg Rich(Me)$

Τώρα μπορούμε να εφαρμόσουμε τον κανόνα της ανάλυσης (κάνοντας κάθε φορά **προτυποποίηση** των μεταβλητών).

Παράδειγμα

Από

$$\neg Rich(Me)$$

και

$$\neg HighlyQualified(z) \vee Rich(z)$$

με MGU $\sigma = \{z/Me\}$, συμπεραίνουμε

$$\neg HighlyQualified(Me).$$

Παράδειγμα

Από

$$\neg Rich(Me)$$

και

$$\neg EarlyEarnings(w) \vee Rich(w)$$

με MGU $\sigma = \{w/Me\}$, συμπεραίνουμε

$$\neg EarlyEarnings(Me).$$

Παράδειγμα

Από

$\neg PhD(x) \vee HighlyQualified(x)$

και

$PhD(y) \vee EarlyEarnings(y)$

με MGU $\sigma = \{x/y\}$, συμπεραίνουμε

$HighlyQualified(y) \vee EarlyEarnings(y)$.

Παράδειγμα

Από

$HighlyQualified(v) \vee EarlyEarnings(v)$

και

$\neg EarlyEarnings(Me)$

με MGU $\sigma = \{v/Me\}$, συμπεραίνουμε

$HighlyQualified(Me)$.

Παράδειγμα

Από

$HighlyQualified(Me)$

και

$\neg HighlyQualified(Me)$

με MGU $\sigma = \{\}$, συμπεραίνουμε τη **κενή φράση.** Επομένως έχουμε καταλήξει σε **αντίφαση**, άρα $KB \models Rich(Me)$.

Παράδειγμα

Θεωρήστε τις φράσεις:

$P(x,x) \vee Q(x) \vee R(x)$

$\neg P(A,z) \vee \neg Q(B)$

Ένας τρόπος να εφαρμόσουμε ανάλυση είναι να διαλέξουμε τα συμπληρωματικά λεκτικά $P(x,x)$ και $P(A,z)$ και με MGU $\sigma = \{x/A, z/A\}$ να βγάλουμε:

$Q(A) \vee R(A) \vee \neg Q(B)$

Παράδειγμα

Ο **δεύτερος τρόπος** να εφαρμόσουμε ανάλυση είναι να διαλέξουμε τα συμπληρωματικά λεκτικά $Q(x)$ και $Q(B)$ και με MGU $\sigma = \{x/B\}$ να βγάλουμε:

$$P(B, B) \vee R(B) \vee \neg P(A, z)$$

Μπορούμε δηλαδή να έχουμε **περισσότερους από ένα τρόπους** για να εφαρμόσουμε τον κανόνα της ανάλυσης σε ένα ζευγάρι φράσεων.

Συζευκτική Κανονική Μορφή

Για να μπορούμε να εφαρμόσουμε ανάλυση, οι δοθέντες τύποι της λογικής πρώτης τάξης πρέπει να είναι σε **συζευκτική κανονική μορφή**.

Ορισμός. Ένας τύπος της λογικής πρώτης τάξης είναι σε **συζευκτική κανονική μορφή (conjunctive normal form, CNF)** αν είναι μια σύζευξη φράσεων.

Μετατροπή σε CNF

1. **Απαλοιφή διπλών και απλών συνεπαγωγών** χρησιμοποιώντας τις παρακάτω ισοδυναμίες:

   $(\phi \Leftrightarrow \psi) \equiv (\phi \Rightarrow \psi \wedge \psi \Rightarrow \phi)$

   $\phi \Rightarrow \psi \equiv \neg\phi \vee \psi$

2. **Μετακίνηση των αρνήσεων ($\neg$) προς τα μέσα** ώστε κάθε άρνηση να εφαρμόζεται σε ένα ατομικό τύπο. Για να το πετύχουμε αυτό, χρησιμοποιούμε τις παρακάτω ισοδυναμίες:

   $\neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi$

   $\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi$

   $\neg(\forall x)\phi \equiv (\exists x)\neg\phi$

   $\neg(\exists x)\phi \equiv (\forall x)\neg\phi$

   $\neg\neg\phi \equiv \phi$

Μετατροπή σε CNF

3. **Προτυποποίηση των μεταβλητών** δηλ. μετονομασία μεταβλητών έτσι ώστε κάθε ποσοδείκτης να δεσμεύει μια διαφορετική μεταβλητή.

4. **Αντικατάσταση υπαρξιακών ποσοδεικτών** ή **μετατροπή κατά Skolem (skolemization).**

   Αν ένας υπαρξιακός ποσοδείκτης δεν βρίσκεται στην εμβέλεια ενός καθολικού ποσοδείκτη, τότε απαλοίφουμε τον ποσοδείκτη και αντικαθιστούμε όλες τις εμφανίσεις της μεταβλητής του ποσοδείκτη με μια νέα σταθερά που λέγεται **σταθερά Skolem**.

   Αν ένας υπαρξιακός ποσοδείκτης $\exists x$ βρίσκεται στην εμβέλεια n καθολικών ποσοδεικτών $\forall y_1, \ldots, \forall y_n$, απαλοίφουμε τον ποσοδείκτη και αντικαθιστούμε όλες τις εμφανίσεις της μεταβλητής του ποσοδείκτη x με τον όρο $F(y_1, \ldots, y_n)$ όπου F είναι ένα νέο σύμβολο συνάρτησης που λέγεται **συνάρτηση Skolem**.

Μετατροπή σε CNF

5. **Απαλοιφή όλων των καθολικών ποσοδεικτών.**
6. **Εφαρμόζουμε την επιμεριστική ιδιότητα του $\vee$ ως προς το $\wedge$:**

   $(\phi \wedge \psi) \vee \theta \equiv (\phi \vee \theta) \wedge (\psi \vee \theta)$

   $\theta \vee (\phi \wedge \psi) \equiv (\theta \vee \phi) \wedge (\theta \vee \psi)$
7. **Απλοποιούμε τις συζεύξεις και διαζεύξεις** απαλοίφοντας τις παρενθέσεις που δεν χρειάζονται.

Παράδειγμα

Ας μετατρέψουμε σε CNF την ακόλουθη πρόταση:

$(\forall x)((\forall y)P(x,y) \Rightarrow \neg(\forall y)(Q(x,y) \Rightarrow R(x,y)))$

1. Απαλοιφή συνεπαγωγών:

   $(\forall x)(\neg(\forall y)P(x,y) \vee \neg(\forall y)(\neg Q(x,y) \vee R(x,y)))$
2. Μετακίνηση $\neg$ προς τα μέσα:

   $(\forall x)((\exists y)\neg P(x,y) \vee (\exists y)(Q(x,y) \wedge \neg R(x,y)))$

Παράδειγμα

3. Προτυποποίηση μεταβλητών:

   $(\forall x)((\exists y)\neg P(x,y) \vee (\exists z)(Q(x,z) \wedge \neg R(x,z)))$

4. Αντικατάσταση υπαρξιακών ποσοδεικτών:

   $(\forall x)(\neg P(x,F_1(x)) \vee (Q(x,F_2(x)) \wedge \neg R(x,F_2(x))))$

5. Απαλοιφή καθολικών ποσοδεικτών:

   $\neg P(x,F_1(x)) \vee (Q(x,F_2(x)) \wedge \neg R(x,F_2(x)))$

6. Επιμερισμός $\vee$ ως προς $\wedge$:

   $(\neg P(x,F_1(x)) \vee Q(x,F_2(x))) \wedge (\neg P(x,F_1(x)) \vee \neg R(x,F_2(x)))$

7. Τελική μορφή:

   $\neg P(x,F_1(x)) \vee Q(x,F_2(x))$

   $\neg P(x,F_1(x)) \vee \neg R(x,F_2(x))$

CNF

Για την περίπτωση που δεν έχουμε Skolemization ισχύει η παρακάτω πρόταση.

Πρόταση. Κάθε τύπος της λογικής πρώτης τάξης είναι ισοδύναμος με ένα τύπο σε CNF.

Αν έχουμε Skolemization, τότε μπορούμε να αποδείξουμε μόνο το παρακάτω.

Πρόταση. Έστω ϕ ένας τύπος της λογικής πρώτης τάξης και ϕ' η μετατροπή του σε CNF. Τότε ϕ είναι ικανοποιήσιμος αν και μόνο αν ϕ' είναι ικανοποιήσιμος.

Παράδειγμα

Θεωρήστε τις εξής φράσεις:

$P(u) \vee P(v)$

$\neg P(x) \vee \neg P(y)$

Οι φράσεις αυτές είναι αντιφατικές (γιατί;) αλλά **δεν** μπορούμε να βγάλουμε την κενή φράση με χρήση ανάλυσης!

Πως μπορούμε να λύσουμε αυτό το πρόβλημα;

Φράσεις: Ένας Ισοδύναμος Ορισμός

Ορισμός. **Φράση** είναι ένα **σύνολο** λεκτικών.

Με τον παραπάνω ορισμό δεν επαναλαμβάνουμε λεκτικά που είναι ίδια όπως κάναμε και στην προτασιακή λογική.

Η έννοια της παραγοντοποίησης που ακολουθεί μας βοηθάει να μην επαναλαμβάνουμε λεκτικά που είναι ισοδύναμα.

Παραγοντοποίηση Φράσεων

Ορισμός. Άν κάποια από τα λεκτικά που αποτελούν τη φράση ϕ ενοποιούνται με MGU γ, τότε η φράση ϕ' που προκύπτει από την εφαρμογή του γ στην ϕ λέγεται **παράγοντας** της ϕ.

Παραδείγματα:

- Η φράση $\{P(F(y)),\ R(F(y),y)\}$ είναι παράγοντας της φράσης $\{P(x),\ P(F(y)),\ R(x,y)\}$. Τα λεκτικά $P(x)$ και $P(F(y))$ ενοποιούνται με MGU $\{x/F(y)\}$.
- Η φράση $\{P(v)\}$ είναι παράγοντας της φράσης $\{P(u), P(v)\}$. Τα λεκτικά $P(u)$ και $P(v)$ ενοποιούνται με MGU $\{u/v\}$).
- Κάθε φράση είναι παράγοντας του εαυτού της. Επίσης, η προτυποποίηση μεταβλητών μπορεί να ερμηνευτεί σαν παραγοντοποίηση.

Αναθεωρημένος Ορισμός της Ανάλυσης

Έστω φράσεις ϕ και ψ. Άν υπάρχει ένα θετικό λεκτικό p σε ένα παράγοντα ϕ' της φράσης ϕ και ένα αρνητικό λεκτικό $\neg q$ σε ένα παράγοντα ψ' της φράσης ψ, και τα λεκτικά p και q ενοποιούνται με MGU γ, τότε με τον **κανόνα της ανάλυσης** μπορούμε να παράγουμε τη φράση:

$$SUBST(\gamma,\ (\phi' \setminus \{p\}) \cup (\psi' \setminus \{\neg q\}))$$

Παράδειγμα

Θεωρήστε τις φράσεις που μας οδήγησαν να αναθεωρήσουμε τον ορισμό της ανάλυσης:

$\{P(u),\ P(v)\}$

$\{\neg P(x),\ \neg P(y)\}$

Για να κάνουμε ανάλυση θεωρούμε τους παράγοντες $\{P(u)\}$ και $\{\neg P(x)\}$ των παραπάνω φράσεων. Τα αντίστοιχα θετικά λεκτικά ενοποιούνται με MGU $\{u/x\}$. Οπότε με χρήση του νέου ορισμού της ανάλυσης παίρνουμε την **κενή φράση.**

Προσοχή: Δεν θα χρησιμοποιήσουμε τον αναθεωρημένο ορισμό της ανάλυσης ξανά στη συνέχεια (ο αρχικός ορισμός αρκεί για τους σκοπούς μας).

Ιδιότητες της Ανάλυσης

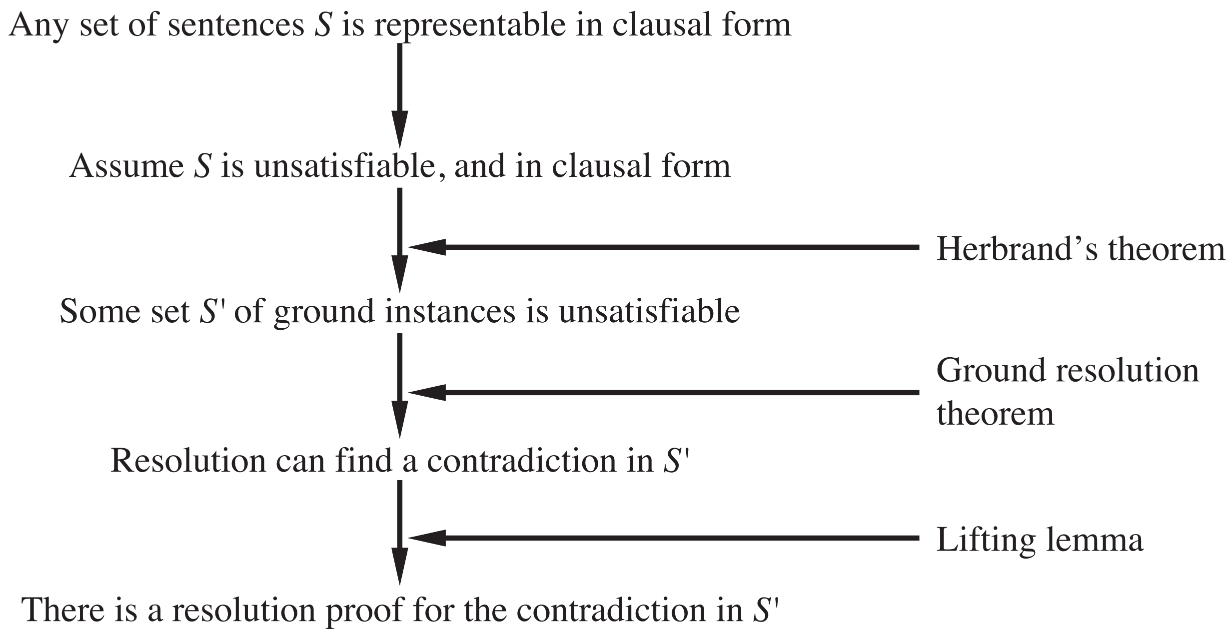
Θεώρημα. (Ορθότητα)
Έστω η βάση γνώσης KB. Αν η ϕ μπορεί να αποδειχθεί από την KB χρησιμοποιώντας ανάλυση, τότε $KB \models \phi$.

Θεώρημα. (Πληρότητα διάψευσης (refutation-completeness))
Αν ένα σύνολο φράσεων Δ είναι μη ικανοποιήσιμο, τότε υπάρχει μια απόδειξη της κενής φράσης με χρήση ανάλυσης από το Δ.

Οι αποδείξεις των θεωρημάτων σκιαγραφούνται στο βιβλίο.

Προσοχή: Η πληρότητα διάψευσης ισχύει μόνο για τον αναθεωρημένο ορισμό της ανάλυσης.

Απόδειξη της Πληρότητας Διάψευσης



Τεχνητή Νοημοσύνη Μ. Κουμπαράκης

Εφαρμογές της Ανάλυσης

Η πιο συνηθισμένη εφαρμογή της ανάλυσης είναι η απόδειξη μια σχέσης **λογικής κάλυψης.**

Αν μας ζητηθεί να αποδείξουμε ότι $KB \models \alpha$ τότε παίρνουμε την άρνηση της α, και δείχνουμε **ισοδύναμα** ότι ο τύπος $KB \land \neg\alpha$ είναι μη ικανοποιήσιμος χρησιμοποιώντας ανάλυση.

Ας δώσουμε τώρα μεγάλο παράδειγμα χρήσης της ανάλυσης, χρησιμοποιώντας το παράδειγμα με τον Συνταγματάρχη West.

Διατύπωση σε Λογική Πρώτης Τάξης

- ... είναι έγκλημα για έναν Αμερικανό να πουλάει όπλα σε εχθρικά κράτη.

  $(\forall x, y, z)\ (American(x) \land Weapon(y) \land Nation(z) \land$

  $Hostile(z) \land Sells(x, z, y) \Rightarrow Criminal(x))$

- Το κράτος Nono ...

  $Nation(Nono)$

- ... που είναι εχθρός της Αμερικής ...

  $Enemy(Nono, America)$

- ... έχει πυραύλους.

  $(\exists x)\ (Owns(Nono, x) \land Missile(x))$

Διατύπωση σε Λογική Πρώτης Τάξης

- Όλους αυτούς τους πυραύλους τους έχει πουλήσει στο Nono ο συνταγματάρχης West ...

  $(\forall x)\ (Owns(Nono, x) \land Missile(x) \Rightarrow Sells(West, Nono, x))$

- ... ο οποίος είναι Αμερικανός.

  $American(West)$

- Η Αμερική είναι κράτος.

  $Nation(America)$

- Οι πύραυλοι είναι όπλα.

  $(\forall x)\ (Missile(x) \Rightarrow Weapon(x))$

- ‘Εχθρός της Αμερικής ’ σημαίνει ‘εχθρικό κράτος ’.

  $(\forall x)\ (Enemy(x, America) \Rightarrow Hostile(x))$

Συζευκτική Κανονική Μορφή

- ... είναι έγκλημα για έναν Αμερικανό να πουλάει όπλα σε εχθρικά κράτη:

  $\neg American(x) \vee \neg Weapon(y) \vee \neg Sells(x, y, z) \vee$

  $\neg Hostile(z) \vee Criminal(x)$

- Το κράτος Nono ...

  $Nation(Nono)$

- ... που είναι εχθρός της Αμερικής ...

  $Enemy(Nono, America)$

- ... έχει πυραύλους.

  $Owns(Nono, M1), \; Missile(M1)$

Συζευκτική Κανονική Μορφή

- Όλους αυτούς τους πυραύλους τους έχει πουλήσει στο Nono ο συνταγματάρχης West ...

  $\neg Missile(x) \vee \neg Owns(Nono, x) \vee Sells(West, x, Nono)$

- ... ο οποίος είναι Αμερικανός.

  $American(West)$

- Η Αμερική είναι κράτος.

  $Nation(America)$

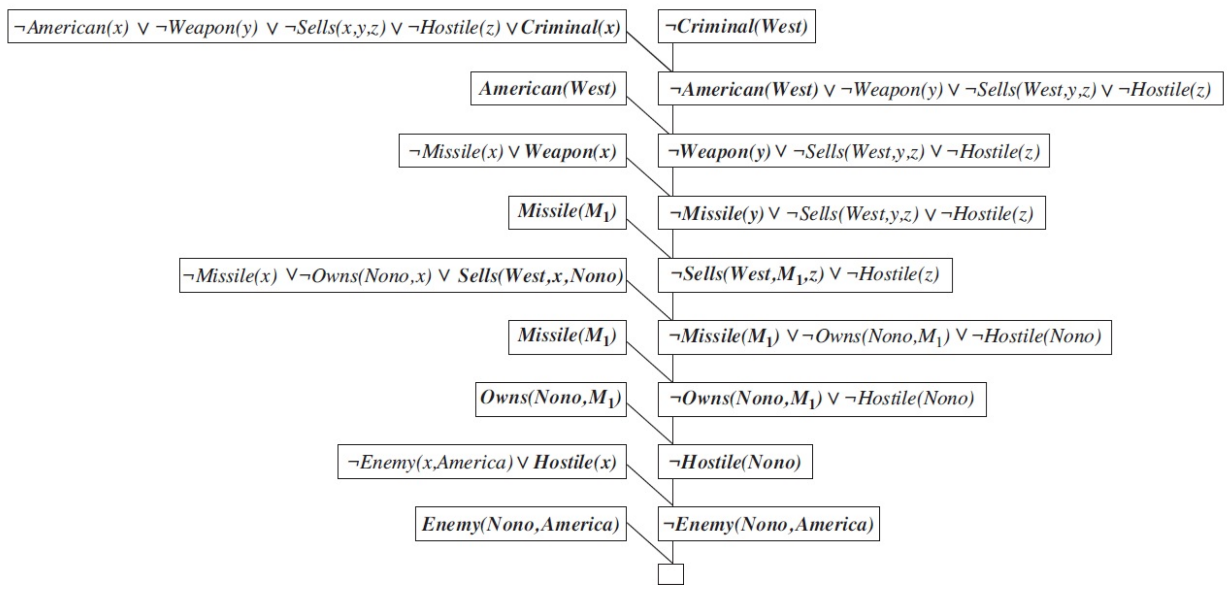
- Οι πύραυλοι είναι όπλα.

  $\neg Missile(x) \vee Weapon(x)$

- ‘Εχθρός της Αμερικής ’ σημαίνει ‘εχθρικό κράτος ’.

  $\neg Enemy(x, America) \vee Hostile(x)$

Απόδειξη



Παράδειγμα

Ας υποθέσουμε ότι γνωρίζουμε τα εξής:

Όλοι όσοι αγαπούν όλα τα ζώα, και αυτούς τους αγαπάει κάποιος.
Κανείς δεν αγαπά όποιον σκοτώνει ζώα.
Ο Jack αγαπάει όλα τα ζώα.
Είτε ο Jack είτε η Περιέργεια σκότωσε τη γάτα που λέγεται Tuna.

Μπορούμε πό τις παραπάνω προτάσεις να αποδείξουμε ότι η Περιέργεια σκότωσε τη γάτα;

Σημείωση: Το παράδειγμα προέρχεται από την παροιμία “Curiosity killed the cat”.

Διατύπωση σε Λογική Πρώτης Τάξης

- Όλοι όσοι αγαπούν όλα τα ζώα, και αυτούς τους αγαπάει κάποιος.

  $(\forall x)((\forall y)(Animal(y) \Rightarrow Loves(x,y)) \Rightarrow (\exists z)Loves(z,x)\ )$

- Κανείς δεν αγαπά όποιον σκοτώνει ζώα.

  $(\forall x)((\exists y)(Animal(y) \wedge Kills(x,y)) \Rightarrow (\forall z)\neg Loves(z,x))$

- Ο Jack αγαπάει όλα τα ζώα.

  $(\forall x)(Animal(x) \Rightarrow Loves(Jack,x))$

Διατύπωση σε Λογική Πρώτης Τάξης

- Είτε ο Jack είτε η Περιέργεια σκότωσε τη γάτα ...

  $Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$

- ... που λέγεται Tuna.

  $Cat(Tuna)$

Επίσης χρειαζόμαστε την πρόταση

$(\forall x)(Cat(x) \Rightarrow Animal(x))$

που αποτελεί **προηγούμενη γνώση (background knowledge).**

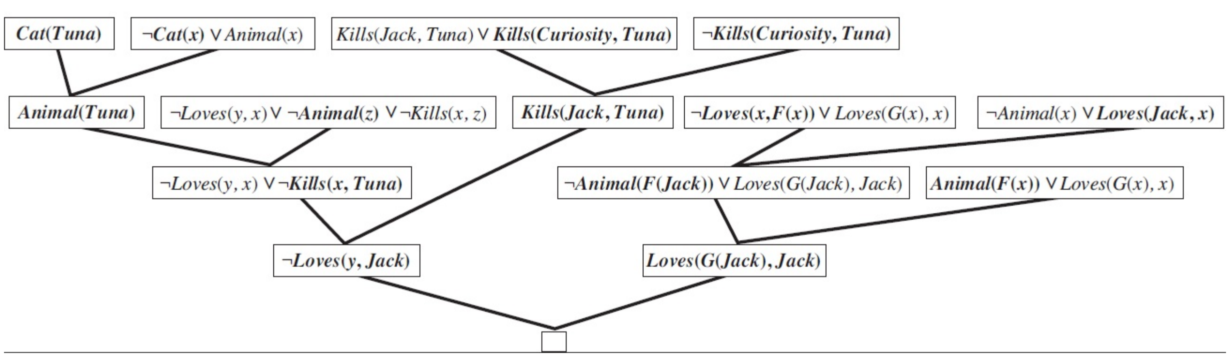
Η άρνηση της πρότασης που θέλουμε να αποδείξουμε είναι:

$\neg Kills(Curiosity, Tuna)$

Συζευκτική Κανονική Μορφή

Κάντε το σαν άσκηση!

Απόδειξη



Ανάλυση, Εγκυρότητα και Μη-ικανοποιησιμότητα

- Πως μπορούμε να χρησιμοποιήσουμε ανάλυση για να αποδείξουμε ότι η παρακάτω πρόταση είναι **έγκυρη**;

  $Happy(John) \lor \neg Happy(John)$

- Πως μπορούμε να χρησιμοποιήσουμε ανάλυση για να αποδείξουμε ότι η παρακάτω πρόταση είναι **μη ικανοποιήσιμη**;

  $Happy(John) \land \neg Happy(John)$

Χρήση της Ανάλυσης για Απάντηση Ερωτημάτων

Μέχρι στιγμής χρησιμοποιήσαμε την ανάλυση για να αποδείξουμε ότι ένας τύπος έπεται από μια βάση γνώσης. Μπορούμε επίσης να χρησιμοποιήσουμε ανάλυση για να **απαντήσουμε ερωτήσεις** σχετικά με γεγονότα που έπονται λογικά από μια βάση γνώσης.

Παράδειγμα: Ποιος σκότωσε την Tuna;

Αυτό το έρωτημα μπορεί να εκφραστεί χρησιμοποιώντας μια ελεύθερη μεταβλητή:

$Kills(x, Tuna)$

Λεκτικά Απάντησης

Ορισμός. Ένα **λεκτικό απάντησης (answer literal)** για ένα ερώτημα ϕ είναι ένας ατομικός τύπος της μορφής $Ans(v_1, \ldots, v_n)$ όπου $v_1, \ldots, v_n$ είναι οι ελεύθερες μεταβλητές του ϕ.

Για να απαντήσουμε το ερώτημα ϕ με χρήση ανάλυσης, σχηματίζουμε τη διάζευξη $Ans(v_1, \ldots, v_n) \vee \neg\phi$ και τη μετατρέπουμε σε CNF.

Μετά χρησιμοποιούμε ανάλυση και τερματίζουμε την αναζητησή μας όταν φτάσουμε σε μια φράση που περιέχει μόνο το λεκτικό απάντησης (αντί την κενή φράση όπως θα κάναμε κανονικά).

Χρήση της Ανάλυσης για Απάντηση Ερωτημάτων

Μπορούμε να διακρίνουμε τις εξής περιπτώσεις:

- **Τερματισμός με μοναδικό λεκτικό απάντησης** $Ans(c_1, \ldots, c_n)$.

  Σε αυτή την περίπτωση, οι σταθερές $c_1, \ldots, c_n$ μας δίνουν μια απάντηση στο ερώτημα. Μπορεί να υπάρχουν περισσότερες απαντήσεις αν υπάρχουν περισσότερες διαψεύσεις ανάλυσης του τύπου $Ans(v_1, \ldots, v_n) \vee \neg\phi$.

  Μπορούμε να συνεχίζουμε να ψάχνουμε περισσότερες απαντήσεις αλλά δεν μπορούμε ποτέ να είμαστε βέβαιοι ότι τις έχουμε βρει όλες (δεν μπορούμε να εγγυηθούμε ότι η ανάλυση θα βρει όλα τα λογικά επακόλουθα ενός συνόλου προτάσεων της λογικής πρώτης τάξης).

Χρήση της Ανάλυσης για Απάντηση Ερωτημάτων

- **Τερματισμός με μια διάζευξη λεκτικών απάντησης.**

  Σε αυτή την περίπτωση, από τη βάση γνώσεων έπεται λογικά μια διάζευξη η οποία δεν μας δίνει μια οριστική απάντηση για το ερώτημα μας.

Παράδειγμα

KB:

$Father(Art, John)$

$Father(Bob, Kim)$

$(\forall x)(\forall y)(Father(x, y) \Rightarrow Parent(x, y))$

Ερώτημα: Ποιος είναι γονιός του John;

Για να απαντήσουμε στο ερώτημα, χρησιμοποιούμε ανάλυση στο παρακάτω σύνολο προτάσεων:

$Father(Art, John)$

$Father(Bob, Kim)$

$\neg Father(x, y) \vee Parent(x, y)$

$Ans(z) \vee \neg Parent(z, John)$

Παράδειγμα

Από

$Father(Art, John)$

και

$\neg Father(x, y) \vee Parent(x, y)$

με MGU $\{x/Art, y/John\}$, έχουμε

$Parent(Art, John)$.

Από τον προηγούμενο τύπο και τον

$Ans(z) \vee \neg Parent(z, John)$

με MGU $\{z/Art\}$, έχουμε

$Ans(Art)$.

Δηλαδή, ο Art είναι γονιός του John.

Παράδειγμα

KB:

$Father(Art, John) \vee Father(Bob, John)$

$(\forall x)(\forall y)(Father(x, y) \Rightarrow Parent(x, y))$

Ερώτημα: Ποιός είναι γονιός του John;

Για να απαντήσουμε στο ερώτημα, χρησιμοποιούμε ανάλυση στο παρακάτω σύνολο προτάσεων:

$Father(Art, John) \vee Father(Bob, John)$

$\neg Father(x, y) \vee Parent(x, y)$

$Ans(z) \vee \neg Parent(z, John)$

Παράδειγμα

Από

$Father(Art, John) \lor Father(Bob, John)$

και

$\neg Father(x, y) \lor Parent(x, y)$

με MGU $\{x/Art, y/John\}$, έχουμε

$Parent(Art, John) \lor Father(Bob, John)$

Από τον παραπάνω τύπο και τον

$\neg Father(x, y) \lor Parent(x, y)$

με MGU $\{x/Bob, y/John\}$, έχουμε

$Parent(Art, John) \lor Parent(Bob, John)$.

Παράδειγμα

Από

$Parent(Art, John) \lor Parent(Bob, John)$

και

$Ans(z) \lor \neg Parent(z, John)$

με MGU $\{z/Art\}$, έχουμε

$Ans(Art) \lor Parent(Bob, John)$

το οποίο στη συνέχεια αναλύεται με

$Ans(z) \lor \neg Parent(z, John)$

με MGU $\{z/Bob\}$, μας δίνει

$Ans(Art) \lor Ans(Bob)$.

Δηλαδή, **ο Art ή ο Bob** είναι γονιός του John.

Η Ανάλυση για Τύπους με Ισότητα

Προσέξτε ότι ο κανόνας της ανάλυσης όπως παρουσιάστηκε, καθώς και τα θεωρήματα ορθότητας και πληρότητας διάψευσης, δεν ισχύουν για φράσεις που περιλαμβάνουν **ισότητα**.

Παράδειγμα: Δεν μπορούμε να χρησιμοποιήσουμε ανάλυση για να συμπεράνουμε $P(B)$ από τη βάση γνώσης

$P(A),\ A = B$

Τι μπορούμε να κάνουμε;

Η Ανάλυση για Τύπους με Ισότητα

Αν οι τύποι της KB περιέχουν το σύμβολο $=$, τότε μπορούμε να κάνουμε ανάλυση με δύο τρόπους:

- Προσθέτουμε στη βάση γνώσης κατάλληλα **αξιώματα** για την σχέση ισότητας.
- Χρησιμοποιούμε, εκτός από την ανάλυση, **ειδικούς κανόνες συμπερασμού** που λαμβάνουν υπόψη τους την ισότητα.
- Χρησιμοποιούμε **ενοποίηση ισότητας (equational unification)**. Η ενοποίηση ισότητας είναι ένα είδος ενοποίησης που λαμβάνει υπόψη του τις σχέσεις ισότητας ανάμεσα στους όρους που ενοποιούνται (δεν είναι δηλαδή απλά συντακτική ενοποίηση).

Το ίδιο ισχύει και για άλλα ειδικά κατηγορήματα όπως τα αριθμητικά $<, \leq$ κλπ.

Τα Αξιώματα της Ισότητας

- Ανακλαστική ιδιότητα: $(\forall x)\ x = x$
- Συμμετρική ιδιότητα: $(\forall x)(\forall y)(x = y \Rightarrow y = x)$
- Μεταβατική ιδιότητα: $(\forall x)(\forall y)(\forall z)(x = y \wedge y = z \Rightarrow x = z)$
- Ιδιότητα της αντικατάστασης ίσων:

  $(\forall x)(\forall y)(x = y \Rightarrow (P_1(x) \Leftrightarrow P_1(y)))$

  $(\forall x)(\forall y)(x = y \Rightarrow (P_2(x) \Leftrightarrow P_2(y)))$

  $\ldots$

  $(\forall w)(\forall x)(\forall y)(\forall z)(w = y \wedge x = z \Rightarrow (F_1(w, x) = F_1(y, z)))$

  $(\forall w)(\forall x)(\forall y)(\forall z)(w = y \wedge x = z \Rightarrow (F_2(w, x) = F_2(y, z)))$

  $\ldots$

  Έχουμε ένα αξίωμα αντικατάστασης ίσων για **κάθε σύμβολο κατηγορήματος και κάθε σύμβολο συνάρτησης.**

Ο Κανόνας της Αποδιαμόρφωσης (Demodulation)

Ο κανόνας συμπερασμού της αποδιαμόρφωσης για την ισότητα είναι ο παρακάτω.

Για οποιουσδήποτε όρους x, y και z που είναι τέτοιοι ώστε $UNIFY(x, z) = \theta$ και οποιοδήποτε λεκτικό $m_n[z]$ που περιέχει τον όρο z:

$$\frac{x = y, \quad m_1 \vee \cdots \vee m_n[z]}{m_1 \vee \cdots \vee m_n[SUBST(\theta, y)]}$$

Παραδείγματα:

- Από $TheThief = John$ και $Arrested(TheThief)$ με MGU $\{\}$, μπορούμε να συμπεράνουμε $Arrested(John)$.
- Από $S(S(0)) = 2$ και $P(2) \vee Q(S(S(w)))$ με MGU $\{w/0\}$, μπορούμε να συμπεράνουμε $P(2) \vee Q(2)$.

Ο Κανόνας της Παραδιαμόρφωσης (Paramodulation)

Ένας άλλος κανόνας συμπερασμού για την ισότητα είναι ο κανόνας της παραδιαμόρφωσης.

Για οποιουσδήποτε όρους x, y και z που είναι τέτοιοι ώστε $UNIFY(x, z) = \theta$ και οποιοδήποτε λεκτικό $m_n[z]$ που περιέχει τον όρο z:

$$\frac{l_1 \vee \cdots l_k \vee x = y, \quad m_1 \vee \cdots \vee m_n[z]}{SUBST(\theta, l_1 \vee \cdots \vee l_k \vee m_1 \vee \cdots \vee m_n[y])}$$

Παράδειγμα: Από τον τύπο

$$P(v) \vee (F(A, v) = F(B, v))$$

και τον

$$Q(B) \vee R(F(A, C))$$

με MGU $\{v/C\}$, μπορούμε να συμπεράνουμε

$$P(C) \vee Q(B) \vee R(F(B, C)).$$

Πληρότητα του Κανόνα της Παραδιαμόρφωσης

Ο κανόνας της παραδιαμόρφωσης είναι πιο γενικός από τον κανόνα της αποδιαμόρφωσης και, μαζί με τον κανόνα της ανάλυσης, έχει ως αποτέλεσμα μια αποδεικτική διαδικασία που είναι **πλήρης ως προς τη διάψευση (refutation complete)** για φράσεις με ισότητα της λογικής πρώτης τάξης.

Η Γλώσσα Prolog

Η Prolog είναι μια γλώσσα λογικού προγραμματισμού. Τα προγράμματα της Prolog στην απλούστερη τους μορφή αποτελούνται από **γεγονότα** ή **κανόνες** (δηλαδή **οριστικές φράσεις Horn**).

Η Γλώσσα Prolog

Θυμίζουμε ότι ένα **γεγονός** είναι μια φράση της μορφής

q

ενώ ένας **κανόνας** είναι μια φράση της μορφής

$\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n \vee q$ (ισοδύναμα $p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q$).

Στην Prolog τα γεγονότα και οι κανόνες γράφονται ως εξής:

```
q.

q:- p1, p2, ..., pn.
```

Prolog

Στην Prolog, μπορούμε επίσης να διατυπώνουμε **ερωτήματα (queries)** χρησιμοποιώντας το εξής συντακτικό:

```
?-p1, p2, ..., pn.
```

Στην Prolog χρησιμοποιείται ο όρος **στόχος (goal)** για να αναφερθούμε σε ένα λεκτικό που περιέχεται σε ένα ερώτημα.

Τα ερωτήματα της Prolog αντιστοιχούν σε φράσεις Horn της μορφής

$$\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n$$

δηλαδή αυτές που αποτελούνται **μόνο από αρνητικά λεκτικά** όπως θα εξηγήσουμε παρακάτω.

Παράδειγμα

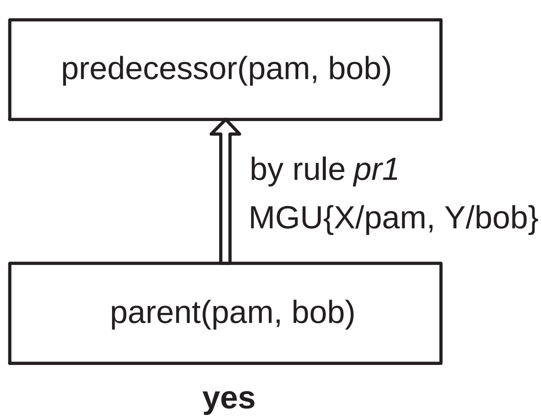
```
predecessor(X, Z) :-    %pr1
  parent(X, Z).
predecessor(X, Z) :-    %pr2
  parent(X, Y),
  predecessor(Y, Z).

parent(pam, bob).   parent(bob, ann).
parent(tom, bob).   parent(bob, pat).
parent(tom, liz).   parent(pat, jim).
```

Ο Κανόνας της Ανάλυσης στην Prolog

- Η Prolog βασίζεται σε μια ειδική μορφή ανάλυσης που λέγεται **SLD-ανάλυση (SLD-resolution)** (Linear resolution with a **S**election function for **D**efinite clauses).
- Σε κάθε βήμα του διερμηνέα της Prolog, εφαρμόζεται ο κανόνας της ανάλυσης ανάμεσα σε ένα στόχο που θέλουμε να αποδείξουμε και ένα γεγονός ή ένα κανόνα του προγράμματος. Έτσι **αποδεικνύεται ο στόχος** (στην περίπτωση του γεγονότος) ή **παράγεται ένα νέο σύνολο στόχων** (στην περίπτωση του κανόνα) που προστίθενται στους στόχους που έχουμε να αποδείξουμε.

Τεχνητή Νοημοσύνη Μ. Κουμπαράκης

Δένδρο Απόδειξης για `?- predecessor(pam,bob)`



Σχόλια για το Δένδρο Απόδειξης

Στην προηγούμενη διαφάνεια, θέλουμε να αποδείξουμε το στόχο

$?- \ predecessor(pam, bob).$

Η χρήση του κανόνα `pr1` από την Prolog είναι ισοδύναμη με την χρήση του κανόνα της ανάλυσης στην άρνηση του υπό απόδειξη στόχου

$\neg predecessor(pam, bob)$

και της φράσης Horn από το πρόγραμμα

$\neg parent(X, Y) \vee predecessor(X, Y)$

ώστε τελικά να παραχθεί το λεκτικό

$\neg parent(pam, bob)$

που είναι η άρνηση του επόμενου υπό απόδειξη στόχου

$?- \ parent(pam, bob)$

Σχόλια για το Δένδρο Απόδειξης

Στο επόμενο βήμα, ο στόχος

$?- \ parent(pam, bob)$

ενοποιείται με το γεγονός

$parent(pam, bob)$

του προγράμματος και η Prolog απαντάει **ναι** στο αρχικό ερώτημα.

Αυτό το βήμα είναι ισοδύναμο με τη χρήση ανάλυσης στο αρνητικό λεκτικό

$\neg parent(pam, bob)$

που είναι η άρνηση του υπό απόδειξη στόχου και στο θετικό λεκτικό

$parent(pam, bob)$

για να φτάσουμε στην **κενή φράση** και το τέλος της απόδειξης.

Σχόλια για το Δένδρο Απόδειξης

Έτσι με βάση την ορθότητα του κανόνα της ανάλυσης, βλέπουμε ότι ο στόχος $predecessor(pam, bob)$ **έπεται λογικά** από το δοσμένο πρόγραμμα Prolog όταν αυτό θεωρηθεί σαν μια βάση γνώσης της λογικής πρώτης τάξης και η Prolog **ορθά** δίνει την απάντηση **ναι** στο ερώτημα.

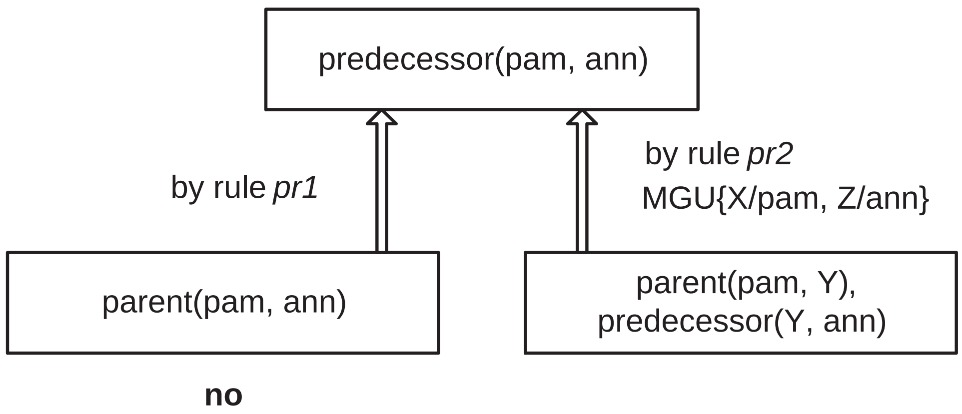
Θέματα Ελέγχου στην Prolog

Δεν θέσαμε το πρόβλημα του ελέγχου (control) που προκύπτει στην λειτουργία του διερμηνέα της Prolog:

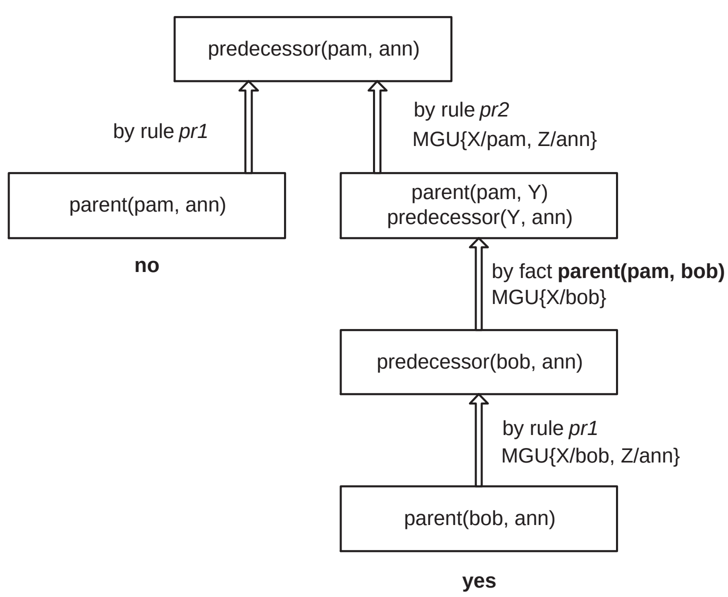
- Με ποιά σειρά διαλέγουμε τον στόχο που θέλουμε να αποδείξουμε όταν έχουμε περισσότερους του ενός στόχους;
- Με ποιά σειρά διαλέγουμε τα γεγονότα που ενοποιούνται με ένα υπό απόδειξη στόχο όταν υπάρχουν περισσότερα του ενός γεγονότα; (ομοίως για κανόνες)

Άσκηση: Εξηγείστε το δένδρο απόδειξης των επόμενων διαφανειών χρησιμοποιώντας ανάλυση.

Δένδρο Απόδειξης του `?- predecessor(pam,ann)`



Δέντρο Απόδειξης του `?- predecessor(pam,ann)`



Υπολογιστική Πολυπλοκότητα της Ανάλυσης

Στη γενική περίπτωση, οι αποδείξεις με χρήση του κανόνα της ανάλυσης μπορούν να μεγαλώνουν **εκθετικά** σύμφωνα με το παρακάτω θεώρημα (ακόμα και στην απλούστερη περίπτωση της προτασιακής λογικής).

Θεώρημα (Haken, 1985). Υπάρχει μια ακολουθία ταυτολογιών $p_1, p_2, p_3, \dots$ της προτασιακής λογικής οι οποίες, όταν μετατραπούν σε μορφή CNF, ο αριθμός συμβόλων καθενός $\neg p_n$ είναι $O(n^3)$, αλλά η μικρότερη διάψευση ανάλυσης του περιέχει τουλάχιστον c^n σύμβολα (για ένα σταθερό $c > 1$).

Εφαρμόζοντας τον Κανόνα της Ανάλυσης Αποδοτικά

Υπάρχουν ποικίλες στρατηγικές που μπορούν να εφαρμοστούν για να κάνουμε την ανάλυση πιο αποδοτική στην πράξη: unit preference, set of support, input resolution, subsumption.

Λεπτομέρειες μπορείτε να βρείτε στο βιβλίο.

Η Ιστορία της Λογικής

450 π.Χ.	Στωϊκή και Μεγαρίτικη σχολή	προτασιακή λογική, πίνακες αληθείας, συμπερασμός (Κανόνας του Θέτειν)
322 π.Χ.	Αριστοτέλης	συλλογισμοί (κανόνες συμπερασμού, ποσοδείκτες)
1847	Boole	προτασιακή λογική
1879	Frege	λογική πρώτης τάξης
1910/13	Whitehead/Russel	Principia Mathematica
1920	Hilbert	'Πρόγραμμα' για την θεμελίωση των Μαθηματικών (δημιουργία ενός τυπικού, αξιωματικού συστήματος για όλα τα Μαθηματικά – απόδειξη ότι το σύστημα αυτό είναι συνεπές, πλήρες και αποφασίσιμο)

Η Ιστορία της Λογικής

1921-22	Post/Wittgenstein	αποδείξεις με πίνακες αληθείας
1930	Gödel	$\exists$ πλήρης μέθοδος συμπερασμού για αποδείξεις της λογικής πρώτης τάξης
1930	Herbrand	πλήρης μέθοδος συμπερασμού για αποδείξεις της λογικής πρώτης τάξης με αναγωγή σε προτασιακή λογική
1931	Gödel	$\neg\exists$ πλήρης μέθοδος συμπερασμού για λογικά συστήματα που περιέχουν την αρχή της επαγωγής (π.χ., αριθμητική)

Η Ιστορία της Λογικής

1936	Turing/Church	Ημι-αποφασισιμότητα του προβλήματος εγκυρότητας για τύπους της λογικής πρώτης τάξης
1960	Davis/Putnam	αποδοτικοί αλγόριθμοι με χρήση ανάλυσης στην προτασιακή λογική
1965	Robinson	η μέθοδος της ανάλυσης για τη λογική πρώτης τάξης

Πληρότητα Συμπερασμού της Λογικής Πρώτης Τάξης

Θεώρημα. (Θεώρημα Πληρότητας του Gödel, 1930)
$KB \models \phi$ ανν $KB \vdash \phi$.

Η σχέση απόδειξης $\vdash$ μπορεί να οριστεί με διάφορους τρόπους. Για παράδειγμα το βιβλίο του Enderton υποθέτει τη χρήση ενός άπειρου συνόλου αξιωμάτων και μόνο τον modus-ponens σαν κανόνα συμπερασμού.

Ισχύει η Υπολογισιμότητα;

Θεώρημα. (Turing/Church, 1936) Το πρόβλημα της λογικής κάλυψης (ισοδύναμα της εγκυρότητας ή της μη ικανοποιησιμότητας) για τη λογική πρώτης τάξης είναι **αναδρομικά αριθμήσιμο (recursively enumerable)**.

Τι σημαίνει όμως αναδρομικά αριθμήσιμο;

Διαισθητικοί Ορισμοί

Ένα πρόβλημα απόφασης (decision problem) P λέγεται **αναδρομικό (recursive)** ή **αποφασίσιμο (decidable)** αν υπάρχει ένας αλγόριθμος τέτοιος ώστε, με είσοδο x, παράγει 'ναι' και τερματίζει όταν $x \in P$, και 'όχι' και τερματίζει όταν $x \notin P$.

Ένα πρόβλημα απόφασης P λέγεται **αναδρομικά αριθμήσιμο (recursively enumerable)** ή **ημιαποφασίσιμο (semi-decidable)** αν υπάρχει ένας αλγόριθμος τέτοιος ώστε, με είσοδο x, παράγει 'ναι' και τερματίζει όταν $x \in P$ αλλά πέφτει σε άπειρο βρόγχο όταν $x \notin P$.

Απόδειξη

Υπάρχουν διάφορες αποδείξεις στη βιβλιογραφία σχετικά με την ημιαποφασισιμότητα του προβλήματος της λογικής κάλυψης για τη λογική πρώτης τάξης.

Αυτό που αξίζει να παρατηρήσουμε είναι ότι χρειαζόμαστε ελάχιστη από την εκφραστική ικανότητα της λογικής πρώτης τάξης για να φτάσουμε σε ημιαποφασισιμότητα π.χ., **μόνο ένα διμελές κατηγόρημα και καθόλου σύμβολα συναρτήσεων.**

Το αποτέλεσμα αυτό είναι από ένα άρθρο των Kalmar and Suranyi στο Journal of Symbolic Logic, Vol. 15, 1950, pp. 161-173.

Θεώρημα Μη-Πληρότητας του Gödel

Θεώρημα. (Gödel, 1930)
Για ένα σύνολο αληθών προτάσεων A της θεωρίας αριθμών και συγκεκριμένα οποιοδήποτε σύνολο από **βασικά αξιώματα** της θεωρίας αριθμών, **υπάρχουν άλλες αληθείς προτάσεις της θεωρίας αριθμών που δεν μπορούν να αποδειχτούν από το** A.

Σημαντικό συμπέρασμα: Οποιοδήποτε μαθηματικό σύστημα που περιέχει την αριθμητική, δεν μπορεί να είναι πλήρες.

Δείτε την περίληψη της ευφυούς απόδειξης του Gödel στο βιβλίο AIMA!

Νεώτερα Αποτελέσματα

Θεώρημα. (Herbrand, 1930)
Αν ένα σύνολο φράσεων Δ είναι μη ικανοποιήσιμο, τότε υπάρχει ένα πεπερασμένο υποσύνολο της βάσης Herbrand του Δ που είναι μη ικανοποιήσιμο.

Θεώρημα. (Robinson, 1965)
Ορθότητα της Ανάλυσης. Αν υπάρχει διάψευση μιας φράσης ϕ από ένα σύνολο φράσεων KB χρησιμοποιώντας ανάλυση, τότε $KB \models \phi$.

Πληρότητα Διάψευσης της Ανάλυσης. Αν ένα σύνολο φράσεων KB είναι μη ικανοποιήσιμο, τότε υπάρχει μια απόδειξη της κενής φράσης από την KB χρησιμοποιώντας ανάλυση.

Τα Όρια των Μεθόδων Απόδειξης

Ερώτηση: Πως μπορούμε να χρησιμοποιήσουμε μια διαδικασία απόδειξης πού είναι πλήρης ως προς τη διάψευση (π.χ. την ανάλυση) για να αποφασίσουμε αν μια πρόταση ϕ έπεται λογικά από ένα σύνολο προτάσεων KB;

Απάντηση: Μπορούμε να πάρουμε την άρνηση της ϕ, να την προσθέσουμε στην KB και να χρησιμοποιήσουμε την αποδεικτική διαδικασία. Αλλά δε θα ξέρουμε αν ισχύει $KB \models \phi$ μέχρι η μέθοδος να βρει μια αντίφαση και να επιστρέψει.

Όσο η μέθοδος δεν έχει επιστρέψει, δεν μπορούμε να ξέρουμε αν η απόδειξη έχει πέσει σε άπειρο βρόγχο ή ότι η απόδειξη πρόκειται να βγει από στιγμή σε στιγμή!

Μερικά Καλά Νέα

Υπάρχουν αρκετά υποσύνολα της λογικής πρώτης τάξης που είναι αποφασίσιμα:

- Η μοναδιαία λογική (monadic logic) (το υποσύνολο της λογικής πρώτης τάξης στο οποίο δεν έχουμε σύμβολα συναρτήσεων και έχουμε μόνο μοναδιαία κατηγορήματα).
- Η Datalog.
- Κάποιες λογικές περιγραφών (description logics) στις οποίες βασίζονται οι μοντέρνες γλώσσες οντολογιών όπως η OWL 2. Δείτε την ιστοσελίδα `http://www.w3.org/TR/2009/REC-owl2-overview-20091027/`.

Πολλά πρακτικά προβλήματα μπορούν να κωδικοποιηθούν με αυτά τα υποσύνολα!

Πράκτορες Βασισμένοι στη Γνώση

function KB-AGENT(*percept*) **returns** an *action*
  static KB, a knowledge-base
    t, a counter, initially 0, indicating time

  TELL(KB,MAKE-PERCEPT-SENTENCE($percept, t$))
  $action \leftarrow$ ASK(KB,MAKE-ACTION-QUERY(t))
  TELL(KB,MAKE-ACTION-SENTENCE($action, t$))
  $t \leftarrow t + 1$
  return *action*

Πως μπορούμε **εύκολα**, χρησιμοποιώντας τους μηχανισμούς συμπερασμού της λογικής πρώτης τάξης που παρουσιάσαμε, να υλοποιήσουμε πράκτορες βασισμένους στη γνώση;

Συστήματα Βασισμένα στη Λογική

- Γλώσσες **λογικού προγραμματισμού (logic programming)** π.χ. Prolog.

  Η Prolog αναπτύχθηκε το 1972 από τον Alain Colmerauer και βασίζεται στην μέθοδο του backward chaning. Η φιλοσοφία της Prolog (σύμφωνα με τον Kowalski, τον μεγάλο θεμελιωτή του λογικού προγραμματισμού) είναι:

  Αλγόριθμος = Λογική + Έλεγχος

  Ο λογικός προγραμματισμός σε Prolog αποτέλεσε τη βάση για περαιτέρω έρευνα και ανάπτυξη στο λογικό προγραμματισμό στις δεκαετίες του 70 και του 80.

Συστήματα Βασισμένα στη Λογική

- Γλώσσες **λογικού προγραμματισμού (logic programming)** π.χ. Prolog.

  Ο λογικός προγραμματισμός και οι προεκτάσεις του εξακολουθούν να είναι χρήσιμες και σήμερα στην έρευνα στο Σημασιολογικό Ιστό, στις Βάσεις Δεδομένων, στις Γλώσσες Προγραμματισμού κλπ.

  Αναφέρουμε επί τη ευκαιρία το **λογικό προγραμματισμό με περιορισμούς (constraint logic programming, CLP)** που ενοποιεί τον κλασσικό λογικό προγραμματισμό με τα προβλήματα ικανοποίησης περιορισμών CSP. Τα συστήματα CLP έχουν χρησιμοποιηθεί με επιτυχία σε πολλές εφαρμογές συνδυαστικής βελτιστοποίησης (π.χ. χρονοπρογραμματισμού, σχεδίασης πλάνων κτλ.)

Συστήματα Βασισμένα στη Λογική

- **Συστήματα παραγωγής (production systems)** που βασίζονται στην μέθοδο forward-chaining (όπου το συμπέρασμα ενός κανόνα/παραγωγής είναι μια ενέργεια που πρέπει να εκτελεστεί).

  Τα συστήματα παραγωγής έχουν χρησιμοποιηθεί αρκετά σε προηγούμενες εφαρμοφές της Τεχνητή Νοημοσύνη (ιδιαίτερα σε expert systems που βασίζονται σε κανόνες).

  Υπάρχουν διάφορα υλοποποιημένα συστήματα παραγωγής όπως το OPS-5 και το CLIPS.

Συστήματα Βασισμένα στη Λογική

- Τα συστήματα **απόδειξης θεωρημάτων (theorem provers)** είναι πιο ισχυρά εργαλεία από την Prolog αφού μπορούν να κάνουν αποδείξεις με οποιουσδήποτε τύπους της λογικής πρώτης τάξης.

  Γνωστά συστήματα: OTTER, PTTP, Prover9 κλπ.

  Η χρήση των συστημάτων απόδειξης θεωρημάτων έχει οδηγήσει σε κάποιες περιπτώσεις **νέα μαθηματικά αποτελέσματα**.

  Επίσης τέτοια συστήματα έχουν χρησιμοποιηθεί και σε άλλες εφαρμογές π.χ., στην επαλήθευση και τη σύνθεση συστημάτων λογισμικού ή υλικού.

Μελέτη

- AIMA, Κεφάλαιο 9.
- M. Genesereth and N. Nilsson. “Logical Foundations of Artificial Intelligence”, Κεφάλαιο 4.

  Αυτό το βιβλίο έχει μια πιο λεπτομερή και τυπική αντιμετώπιση των θεμάτων που παρουσιάσαμε.

Μελέτη

- Donald MacKenzie. Mechanizing Proof: Computing, Risk and Trust. MIT Press, 2001.

  Μια ιστορική και κοινωνιολογική μελέτη του προβλήματος της αυτοματοποιημένης απόδειξης θεωρημάτων. Διαβάστε το αν έχετε χρόνο!
- A. Δοξιάδης, X. Παπαδημητρίου, A. Παπαδάτος και A. Di Donna. Logicomix. Εκδόσεις Ίκαρος 2008.

  Η ιστορία της λογικής σε κόμιξ.