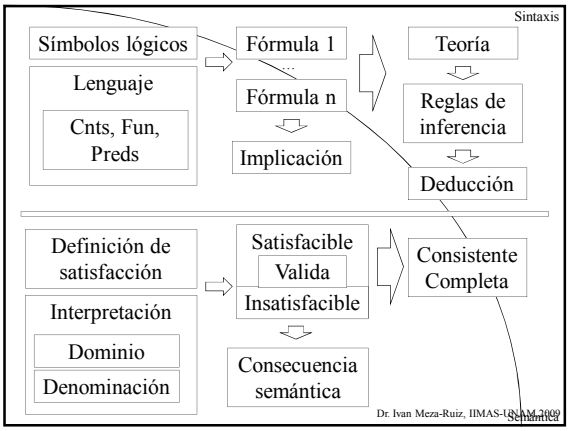
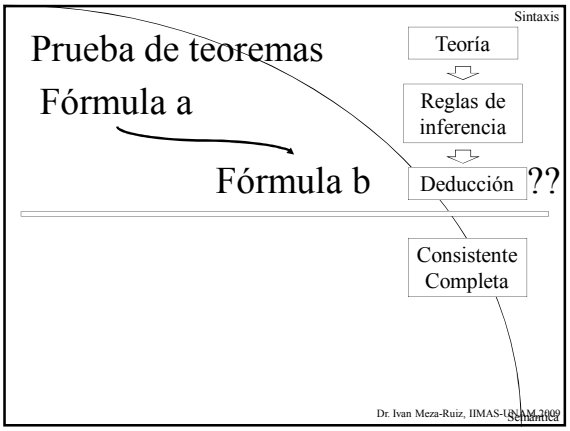
Razonamiento automático

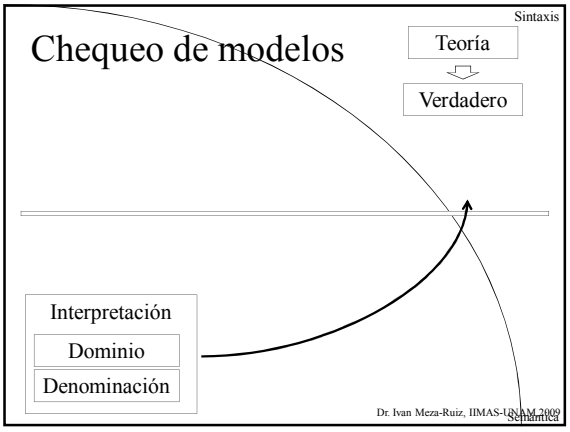
Basado en: Logic in Computer Science, Hunt & Ryan

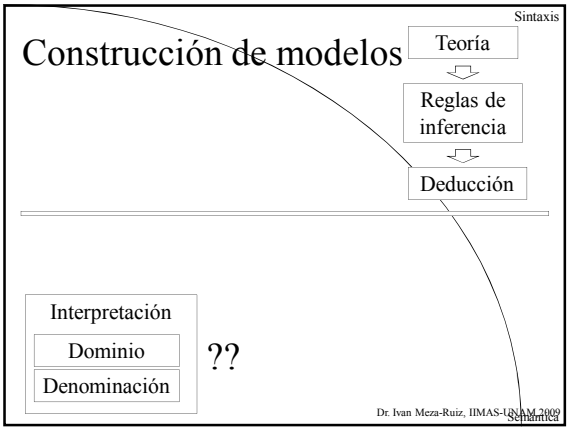
Técnicas

- Prueba de teoremas (Theorem provers)
- Chequeo de modelos (Model checkers)
- Constructores de modelos (Model builders)









Deducción natural

- Un conjunto de reglas de inferencia para lógica proposicional
  - Conjunción
  - Doble negación
  - Implicación
  - Negación

Lógica proposicional

- Sintaxis
  - ¬ & ∨ → ↔ ( )
  - Variables booleanas
- Propiedades
  - Es completa
  - Es consistente

Conjunción

$$\frac{\alpha \qquad \beta}{\alpha \,\&\, \beta}$$ Introducción

$$\frac{\alpha \,\&\, \beta}{\alpha} \qquad \frac{\alpha \,\&\, \beta}{\beta}$$ Eliminación

Doble negación

$$\frac{\alpha}{\neg\neg\alpha}$$ Introducción

$$\frac{\neg\neg\alpha}{\alpha}$$ Eliminación

Implicación

$$\frac{\begin{bmatrix}\alpha \\ \vdots \\ \beta\end{bmatrix}}{\alpha \rightarrow \beta}$$ Introducción

$$\frac{\alpha \qquad \alpha \rightarrow \beta}{\beta}$$ Eliminación
Modus ponens

Modus tollens

$$\frac{\alpha \rightarrow \beta \qquad \neg\beta}{\neg\alpha}$$ Eliminación
Modus tollens

Disyunción

$$\frac{\alpha}{\alpha \vee \beta} \qquad \frac{\beta}{\alpha \vee \beta}$$

Introducción

$$\frac{\alpha \vee \beta \quad \begin{bmatrix}\alpha \\ \vdots \\ \chi\end{bmatrix} \quad \begin{bmatrix}\beta \\ \vdots \\ \chi\end{bmatrix}}{\chi}$$

Eliminación

Contradición

$$\frac{\begin{bmatrix}\alpha \\ \vdots \\ \bot\end{bmatrix}}{\neg\alpha}$$

Introducción

$$\frac{\bot}{\alpha} \qquad \frac{\alpha \quad \neg\alpha}{\bot}$$

Eliminación

Deducción

1			p & q → r	Premisa
	2		p	Asumción
		3	q	Asumción
		4	p & q	&i 2,3
		5	r	→e 1,4
	6		q→r	→i 3-5
7			p→(q→r)	→i 2-6

El reto es…

1			p & q → r	Premisa
…			…	…
m			p→(q→r)	…

Formas normales

- Permiten evaluar rápidamente la validez de una fórmula:
  - q v a v c v ¬q
- CNF (Conjunctive Normal Form)
  - Literales: átomo o negación
  - CNF formula: conjunción de clausulas formadas por disyunción de literales
    - L ::= p | ¬p
    - D ::= L | L v D
    - C ::= D | D & C

De fórmula prop. a CNF

- Eliminar las implicaciones
  - p → q = ¬ p v q
- Transformar a Negation Normal Form
  - ¬ ¬ p =p
  - ¬(p & q) = ¬ p v ¬q
  - ¬(p v q) = ¬ p & ¬q

De fórmula prop. a CNF (cont.)

Def CNF(ϕ)

If es_literal(ϕ) return ϕ

If es_con(ϕ) return CNF(fst(ϕ)) & CNF(snd(ϕ))

If es_dis(ϕ) return DISTR(CNF(fst(ϕ)),CNF(snd(ϕ)))

De fórmula prop. a CNF (cont.)

- DISTR usa la equivalencia distributiva para eliminar conjunciones dentro de disyunciones
  - p1 & p2 v q = p1vq & p2 v q

Ejemplo

$$\texttt{CNF}\,(\texttt{NNF}\,(\texttt{IMPL\_FREE}\,(\neg p \wedge q \rightarrow p \wedge (r \rightarrow q))))$$

$$\begin{aligned}
\texttt{IMPL_FREE}\,(\phi) &= \neg\texttt{IMPL_FREE}\,(\neg p \wedge q) \vee \texttt{IMPL_FREE}\,(p \wedge (r \rightarrow q)) \\
&= \neg((\texttt{IMPL_FREE}\,\neg p) \wedge (\texttt{IMPL_FREE}\,q)) \vee \texttt{IMPL_FREE}\,(p \wedge (r \rightarrow q)) \\
&= \neg((\neg p) \wedge \texttt{IMPL_FREE}\,q) \vee \texttt{IMPL_FREE}\,(p \wedge (r \rightarrow q)) \\
&= \neg(\neg p \wedge q) \vee \texttt{IMPL_FREE}\,(p \wedge (r \rightarrow q)) \\
&= \neg(\neg p \wedge q) \vee ((\texttt{IMPL_FREE}\,p) \wedge \texttt{IMPL_FREE}\,(r \rightarrow q)) \\
&= \neg(\neg p \wedge q) \vee (p \wedge \texttt{IMPL_FREE}\,(r \rightarrow q)) \\
&= \neg(\neg p \wedge q) \vee (p \wedge (\neg(\texttt{IMPL_FREE}\,r) \vee (\texttt{IMPL_FREE}\,q))) \\
&= \neg(\neg p \wedge q) \vee (p \wedge (\neg r \vee (\texttt{IMPL_FREE}\,q))) \\
&= \neg(\neg p \wedge q) \vee (p \wedge (\neg r \vee q)).
\end{aligned}$$

Ejemplo (cont.)

$$\begin{aligned}
\texttt{NNF}\,(\texttt{IMPL_FREE}\,\phi) &= \texttt{NNF}\,(\neg(\neg p \wedge q)) \vee \texttt{NNF}\,(p \wedge (\neg r \vee q)) \\
&= \texttt{NNF}\,(\neg(\neg p) \vee \neg q) \vee \texttt{NNF}\,(p \wedge (\neg r \vee q)) \\
&= (\texttt{NNF}\,(\neg\neg p)) \vee (\texttt{NNF}\,(\neg q)) \vee \texttt{NNF}\,(p \wedge (\neg r \vee q)) \\
&= (p \vee (\texttt{NNF}\,(\neg q))) \vee \texttt{NNF}\,(p \wedge (\neg r \vee q)) \\
&= (p \vee \neg q) \vee \texttt{NNF}\,(p \wedge (\neg r \vee q)) \\
&= (p \vee \neg q) \vee ((\texttt{NNF}\,p) \wedge (\texttt{NNF}\,(\neg r \vee q))) \\
&= (p \vee \neg q) \vee (p \wedge (\texttt{NNF}\,(\neg r \vee q))) \\
&= (p \vee \neg q) \vee (p \wedge ((\texttt{NNF}\,(\neg r)) \vee (\texttt{NNF}\,q))) \\
&= (p \vee \neg q) \vee (p \wedge (\neg r \vee (\texttt{NNF}\,q))) \\
&= (p \vee \neg q) \vee (p \wedge (\neg r \vee q)).
\end{aligned}$$

Ejemplo (cont. 2)

$$\begin{aligned}
\texttt{CNF}\,(\texttt{NNF}\,(\texttt{IMPL_FREE}\,\phi)) &= \texttt{CNF}\,((p \vee \neg q) \vee (p \wedge (\neg r \vee q))) \\
&= \texttt{DISTR}\,(\texttt{CNF}\,(p \vee \neg q), \texttt{CNF}\,(p \wedge (\neg r \vee q))) \\
&= \texttt{DISTR}\,(p \vee \neg q, \texttt{CNF}\,(p \wedge (\neg r \vee q))) \\
&= \texttt{DISTR}\,(p \vee \neg q, p \wedge (\neg r \vee q)) \\
&= \texttt{DISTR}\,(p \vee \neg q, p) \wedge \texttt{DISTR}\,(p \vee \neg q, \neg r \vee q) \\
&= (p \vee \neg q \vee p) \wedge \texttt{DISTR}\,(p \vee \neg q, \neg r \vee q) \\
&= (p \vee \neg q \vee p) \wedge (p \vee \neg q \vee \neg r \vee q)\,.
\end{aligned}$$

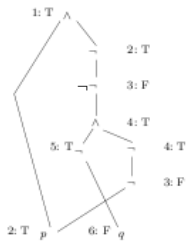
Satisfacible, Válida y Insatisfacible

- Si α es válida es satisfacible
- Si α es satisfacible no es insatisfacible
- Si α es válida entonces ¬α es insatisfacible
- Si ¬α es valida entonces α insatisfacible
- Si ¬α es in satisfacible entonces α es válida

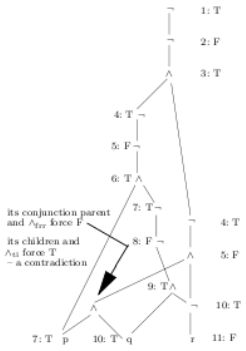
SAT solvers

- Tratan de asignar a las variables de una lógica proporcional valores de falso y verdadero que hagan a la fórmula verdadera
- Es decir si tienen existo muestran que la fórmula es satisfacible, caso contrario insatisfacibles.

Linear solver

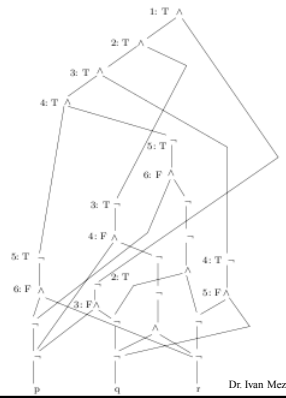


p & ¬¬(¬q & ¬¬p)



¬(p & q)

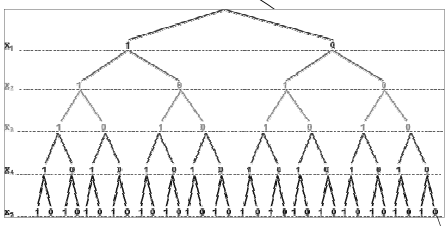
- Nuestro método falla para estos casos



Salidas de este método

- Se encuentra una asignación en todos los nodos: satisfacible
- Se encuentra que una asignación es contradictoria con una previa: insatisfacible
- No todos los nodos tienen asignación:
  - ☹

Otra alternativa



$f = (\neg x_2 \vee x_5) \wedge (x_1 \vee \neg x_3 \vee x_4) \wedge (x_4 \vee \neg x_5) \wedge (x_1 \vee x_2)$

Davis Putnam Loveland

- Obtener la CNF de la fórmula
- Escoger una variable
- Asignar una valor de verdad
- Para las clausulas donde la variable o su negación son verdaderas eliminar la clausula
- Para las clausulas donde la variable o su negación son falsas eliminar la variable

Métodos incompletos

- GSAT y WalkSat
- Implementan una estrategia de búsqueda y reducción de espacio de búsqueda
- Utilizan un aspecto aleatorio

- SAT es NP-Complete para K>3
- MAX-SAT es NP-hard

Estado del arte en SATs

- http://www.satcompetition.org/2009/

Recapitulando

- Lógica proposicional
  - Opción uno: manipular fórmulas con reglas de inferencia, ejemplo: reglas de deducción natural: Theorem Provers
  - Opción dos: asignar una interpretación para identificar si fórmula es satisfacible: SAT solvers

Lógica de primer orden

- Sintaxis
  - ¬ & ∨ → ↔ ∀ ∃ = ( ) ,
  - Constantes, predicados, funciones
- Propiedades
  - Indecidible, en algunos casos no podemos determinar si una formula es valida, ni siquiera insatisfacible
  - LPO no es completa o consistente al mismo tiempo para lógicas que incluyan la teoría de los números naturales

Cuantificador universal

$$\frac{\begin{bmatrix} x_0 \\ \vdots \\ \alpha\{x_0/\tau\} \end{bmatrix}}{\forall x \alpha}$$

Introducción

$$\frac{\forall x \alpha}{\alpha\{x/\tau\}}$$

Eliminación

Cuantificador existencial

$$\frac{\alpha\{x/\tau\}}{\exists x \alpha}$$

Introducción

$$\frac{\exists x \alpha \quad \begin{bmatrix} x_0 & \alpha\{x_0/\tau\} \\ & \vdots \\ & \chi \end{bmatrix}}{\chi}$$

Eliminación

Deducción

1				∃xP(x)	Premisa
2				∀x∀y(P(x)→Q(y))	Premisa
	3		y0		
		4	x0	P(x0)	Asumción
		5		∀y(P(x0)→Q(y))	∀x e 2
		6		P(x0)→Q(y0)	∀y e 5
		7		Q(y0)	→i 6,4
	8			Q(y0)	∃x e 1,4-7
7				∀yQ(y)	∀y i 3-8

El reto

1				∃P(x)	Premisa
2				∀x∀y(P(x)→Q(y))	Premisa
…				…	…
7				∀yQ(y)	…

¿Dónde estamos?

- FOL es más expresivo
- Pero, FOL, agrega aspectos de indecidebilidad

Prover9

- Prover9 es un TP
- Recibe una teoría y un objetivo y llena la parte de en medio.
- Todo es manipulación sintáctica

Estado del arte TP

- http://www.cs.miami.edu/~tptp/CASC/22/
- http://www.cs.miami.edu/~tptp/

Otra opción

- SAT solvers manipulan una formula tratando de aplicar una interpretación.
- TP manipulan una teoría para identificar que sigue de ellas para, generalmente siguen un objetivo.
- Qué tal si tengo una interpretación y checo si la formula la satisface.
- Como estamos en FOL la interpretación es más compleja que una tabla de verdad

Verificación

- Verificar que un modelo cumpla con una especificación (conjunto de propiedades).

Criterios verificación

- Basados en pruebas: Especificación (fórmula) sigue a sistema (teoría)
- Basados en modelos: Sistema (FSM) satisface a especificación (fórmula)
- Automático: Automático o interactivo?
- Alcance: propiedades o sistemas completos
- Dominio: software o hardware
- Pre vs pos desarrollo

Chequeo de modelos

- Automático
- Basados en modelos
- Verifica propiedades
- Técnicas
  - Buscar contra ejemplos
  - Partir de la especificación y checar si son válidos

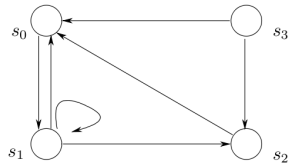
Aplicaciones

- Dado un modelo checar que la fórmula se satisfaga
  - SQL
  - Xqueries
- ¿Qué tal modelos de software, estándares de diseño y modelos de ejecución?
  - Se abstraen usando gráficas dirigidas
  - Formula que representa ir de un estado bueno a uno malo

Gráficas dirigidas

- Fórmulas para describir propiedades deseables de las gráficas requieren de fórmulas variables en FOL
  - Constantes: Nodos
  - Predicados: Links
  - Fórmula de paths:
    - Caso el nodo es el mismo
    - Caso hay un nodo en medio
    - Caso hay dos nodos en medio
    - etc

Ejemplo



Lógica de Segundo Orden

- $\exists P\alpha$
- $\forall P\alpha$
- LSO permite definir paths para gráficas dirigidas
- ¿Qué hay de Lógica de tercer orden?
  - HOL usadas extensivamente en TP

¿Qué tan importante es?

- Clarke, Emerson, and Sifakis ganaron el Turing Award 2007
- Busca responder:
  - M, s $\models \alpha$

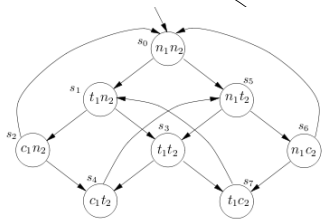
Lógica para MC

- Usa lógica temporal: HOL
- Busca responder:
  - M $\models \alpha$
- Se modela el sistema para crear el modelo M
- Se codifica la especificación/propiedades usando HOL
- Se ejecuta el sistema

Lógica temporal lineal LTL

- El valor de verdad depende del tiempo
- L. propociosional más nuevos conectivos para representar
  - Siguiente estado (N)
  - Algún futuro estado (F)
  - Todos los estados (G)
  - Hasta el estado (U)
  - Otros…

Ejemplo



n – Estado no crítico
t – Estado tratando de entrar en estado critico
c – En estado crítico

Especificación

- Propiedades
  - Solamente un proceso está en estado crítico
  - Si un proceso necesita entrar en su estado crítico debe lograrlo eventualmente
  - Un proceso puede pedir en estado crítico
  - No existe una secuencia definida para entrar àl estado crítico

Especificación en LTL

- G¬(c1 & c2)
- G(t1 → Fc1)
- No se puede, no existen
- No se puede, pero complemento sí:
  - G(c1 →c1 W(¬c1 & ¬ c1 W c2))

Estado del arte en MC

- http://nusmv.fbk.eu/

Model building

- ¿Qué pasa si no tengo el modelo/interpretaciòn?
- Model building busca crear una interpretación para una especificación dada.

Mace4

- Mace4 es un model builder que viene con Prover9

En la práctica

- En entendimiento del habla
  - Se tiene un discurso (Teoría y modelo)
  - Se tiene una oración
  - Se obtiene una representación semántica (Fórmulas)
  - Se prueba que formula sigue a la teoría (TP)

En la práctica (cont.)

- TP busca satisfacibilidad puede que nunca regrese
  - Se busca que formula regresa modelo para teoria+formula+modelo inicial (MC)
- Se utilizan ambos en paralelo, el primero en regresar nos da una respuesta
- Si ninguno regresa, se considera a la fórmula incoherente