Resumen

Este documento junta todos los entregables pedidos por el profesor en un pdf

Índice

1. Reporte =Documentación

Campus: Ciudad Universitaria
Facultad: Ingeniería
Materia : Inteligencia Artificial
Semestre: 2022-2
Equipo: 1
Clave: 0406
Participantes:
- Barrera Peña Víctor Miguel
- Espino De Horta Joaquín Gustavo

Profesor: Dr. Ismael Everardo Barcenas Patiño
Título : Proyecto 2
Subtítulo : Fórmula de lógica proposicional valida o no

Figure 7

$$\dfrac{\dfrac{f^{\omega} \to_{\varepsilon} g(f^{\omega})}{f^{\omega} \to^{\infty} g(f^{\omega})}}{f^{\omega} \to^{\infty} f(g(f^{\omega}))} \quad f(g(f^{\omega})) \to_{\varepsilon} g(g(f^{\omega})) \quad \dfrac{\dfrac{\dfrac{f^{\omega} \to^{\infty} g^{\omega}}{g(f^{\omega}) \to^{\infty} g^{\omega}}}{g(f^{\omega}) \to^{\infty} g^{\omega}}}{g(g(f^{\omega})) \to^{\infty} g^{\omega}}$$

$$\dfrac{f^{\omega} \to^{\infty} g^{\omega}}{g(f^{\omega}) \to^{\infty} g^{\omega}}$$

$$\dfrac{f^{\omega} \to_{\varepsilon} g(f^{\omega}) \qquad \dfrac{f^{\omega} \to^{\infty} g^{\omega}}{g(f^{\omega}) \to^{\infty} g^{\omega}}}{f^{\omega} \to^{\infty} g^{\omega}}$$

Índice

Definición del problema

Escribir un programa lógico que determine si una fórmula de lógica proposicional es válida o no.

• Algoritmo basado en el cálculo de secuentes. • Entrada: una expresión de lógica proposicional. • Salida: una prueba (en código latex), si existe, de lo contrario, no válida.

Capítulo 0 Estructura del repositorio

El repositorio documenta todo el proceso hecho para realizar el programa , incluyendo los pdf´s

Bibliotecas usadas

```
\usespackage{prof}
\usespackage{bussproofs}
```

Para poder ejecutar el programa es necesario tener instalado un interprete de `prolog` para poder ejecutar el programa.

Para dar la apariencia a las etiquetas de los `prooftree` pues se uso

```
\mathcal{I} \mathcal{D}
```

Capítulo 1 Introducción

"Aquel modo de razonamiento que examina el estado de las cosas afirmadas en las premisas, forma un diagrama de este estado de cosas, percibe en las partes de ese diagrama relaciones no mencionadas expl´ıcitamente en las premisas, se satisface a s´ı mismo mediante experimentos mentales sobre el diagrama, de que estas relaciones siempre subsistir´an, y concluye su verdad necesaria o probable" Charles Sanders Peirce.

Razonar partiendo de premisas mediante reglas para llegar a conclusiones es la tarea que tomamos desde la matemática para poder demostrar, este es uno de los principales objetivos de la lógica. Formalmente se demuestra la validez de un razonamiento mediante el método de deducción.

El método de la deducción se hace usando las leyes de lógica de primer orden o mejor conocidas como **deducción natural**. Para logar obtener un resultado se hace uso de su principal herramienta que es el **cálculo de secuentes**.

"El cálculo de secuentes presenta numerosas analogía con la deducción natural y, aunque se adapta bien al caso intuicionista, también resuelve problemas de validez y consecuencias lógicas para la lógica proposicional clásica."

Capítulo 2 Desarrollo

Idea de desarrollo del programa

Casos de prueba

Triviales (1 caso)

$$\frac{}{A \vdash A}\ \mathcal{ID}$$

Fáciles (3 casos)

Caso 1

$$\dfrac{\dfrac{\dfrac{}{\vdash \neg A, A}\ \mathcal{ID}}{\vdash \neg A, A}\ \mathcal{D}\neg}{\neg\neg A \vdash A}\ \mathcal{I}\neg$$

Caso 2

$$\dfrac{\dfrac{}{A \vdash A}\ \mathcal{ID} \qquad \dfrac{}{B \vdash B}\ \mathcal{ID}}{A, B \vdash A \wedge B}\ \mathrm{R}$$

Caso 3

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\ \mathcal{ID}}{\vdash \neg A, A}\ \mathcal{R}\neg}{\vdash A, \neg A}\ \mathcal{RX}}{\vdash A \vee \neg A}\ \mathcal{ANT}$$

Media (3 casos)

Caso 1

$$\dfrac{\dfrac{\dfrac{B \vdash A, A \qquad B, B, \vdash A}{B, A \to B \vdash A}\ \mathcal{I}\to}{A \to B \vdash B \to A}\ \mathcal{D}\to \qquad \dfrac{\dfrac{A \vdash B, B \qquad A, A \vdash B}{A, B \to A \vdash B}\ \mathcal{I}\to}{B \to A \vdash A \to B}\ \mathcal{D}\to}{\vdash (A \vee B) \to (A \leftrightarrow B)}\ D\leftrightarrow$$

Caso 2

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}\ \mathcal{ID} \qquad A \vdash B}{A \vdash (A \wedge B)}\ \mathcal{D}\wedge \qquad \dfrac{\dfrac{\dfrac{}{A \vdash A}\ \mathcal{ID}}{A, B \vdash A}\ \mathcal{I} - Debilitamiento}{(A \wedge B) \vdash A}\ \mathcal{I}\wedge}{\vdash A \leftrightarrow (A \wedge B)}\ \mathcal{D}\leftrightarrow$$

Caso 3

$$\dfrac{\dfrac{\dfrac{\vdash A, B}{\neg A \vdash B}\,\mathcal{J}\neg}{\vdash (A \vee \neg B)}\,\mathcal{D}\vee \qquad \dfrac{\dfrac{A, B \vdash}{A \vdash \neg B}\,\mathcal{D}\neg}{\vdash (A \vee \neg B)}\,\mathcal{D}\vee}{\vdash (\neg A \vee B) \wedge (A \vee \neg B)}\,\mathcal{D}\ \textit{Conjugacion}$$

Difíciles (3 casos)

Caso 1 (revisar)

$$\dfrac{\dfrac{\dfrac{\vdash A, B}{\neg A \vdash B}\,\mathcal{J}\neg}{\vdash (A \vee \neg B)}\,\mathcal{D}\vee \qquad \dfrac{\dfrac{A \vdash A}{A, A \to B \vdash B}\,\mathcal{D}\neg}{A \to \vdash A \to B}\,\mathcal{D}\vee}{\vdash (A \vee B) \to (A \leftrightarrow B)}\,\mathcal{D}\to$$

Caso 2 Caso 3

Sin solución (1 caso)

$$\dfrac{}{A \vdash B}\,\mathcal{J}\mathcal{D}$$

Caso 1

Código

```
//Definir operaciones//

:-op(1,fx,neg).
:-op(2,xfy,or).
:-op(2,xfy,and).
:-op(2,xfy,imp).
:-op(2,xfy,dimp).

//Win Condition
sq(G,D):-
sq([F],[F]).

//Debilitamiento
sq([F|G],D):- atom(F),
sq(G,D).

sq(G,[F|D]):- atom(F),
sq(G,D).
```

```
//Negacion
sq([neg F|G],D):-
sq(G,[F|D]),
//write(Archivo,"\neg"+G).

sq(G,[neg F|D]):-
sq([F|G],D).

//Disyuncion
sq([F or R|G],D):-
sq([F|G],D),
sq([R|G],D).

sq(G,[F or R|D]):-
append([F,R],D,U),
sq(G,U).

//Conjuncion
sq([F and R|G],D):-
append([F,R],G,U),
sq(U,D).

sq(G,[F and R|D]):-
sq(G,[F|D]),
sq(G,[R|D]).

//Implicacion
sq([F imp R|G],D):-
sq(G,[F|D]),
sq([R|G],D).

sq(G,[F imp R|D]):-
sq([F|G],[R|D]).

//Doble Implicacion
sq([F dimp R|G],D):-
append([F,R],D,U),
sq(G,U),
append([F,R],G,U),
sq(U,D).

sq(G,[F dimp R|D]):-
sq([F|G],[R|D]),
sq([R|G],[F|D]).
```

Explicación código

Capítulo 3 Conclusión

Barrera Peña Víctor Miguel

Crear un programa en un nuevo paradigma de programación es un reto y más si aquel lenguaje haz hecho pocos ejercicios, además de entrar en conflicto con latex para crear las ecuaciones constituye un reto de buen nivel. Por anteriores motivos puedo decir que si bien hasta el momento no se ha logrado obtener todo a la perfección nos hemos acercado al objetivo de crear un programa que pueda crear los secuentes usando programación lógica.

En lo que respecta, posiblemente con una semana más y un poco de práctica para encontrar el error de prolog es suficiente para poder concluir el proyecto con exito.

Espino de Horta Joaquín Gustavo

El mayor desafío de este proyecto sin duda era el conocer y manipular correctamente el lenguaje argumentativo de Prolog un programa cuya única función es la de comprobar hechos y operaciones lógicas. Si bien el programa a realizar es en cierto punto sencillo, no se podía realizar de la manera correcta por muchos factores de falta de información, donde no se encuentra mucho soporte o documentación adecuada así como ciertas funciones que nos acortaban trabajo.

Apuesto que si nos hubiesen dejado este mismo trabajo, con la gran libertad de hacerlo con un lenguaje de programación de libre elección, se habría llegado a un producto satisfactorio en menos de una semana como si ocurrió con el proyecto anterior. En cuanto al objetivo, cumple en el ámbito teórico así como su propuesta de ser posible a realizar en las propiedades de Prolog.

Esperamos en los siguientes proyectos poder afrontarlos con la libertad necesaria de herramientas o con un mayor conocimiento de las condiciones a las que estaremos sujetos. Pues, lo que puedo argumentar, sería realmente fácil este proyecto como lo sugiere el sentido común, aún con la generación de archivo Latex si se permitiera usar cualquier lenguaje.

Anexos para investigación

Charles Sanders Peirce

- La ciencia de la semiótica,http://mastor.cl/blog/wp-content/uploads/2015/08/PEIRCE-CH.-S.-La-Ciencia-de-La-Semi %C3 %B3tica.pdf

Fuentes

- FUENTES GUZMAN, D. A. N. I. E. L. C. A. M. I. L. O. (2014). CÁLCULO DE SECUENTES Y GRAFICOS EXISTENCIALES ALFA: DOS ESTRUCTURAS EQUIVALENTES PARA LA LOGICA PROPOSICIONAL. UNIVERSIDAD DEL TOLIMA. Recuperado 7 de abril de 2022, de http://repository.ut.edu.co/bitstream/001/1172/1/RIUT-ABA-spa-2014-C %C3 %A1lculo %20de %20secuentes %20y %20gr %C3 %A1ficos %20existenciales %20Alfa. %20 %20Dos

2. Diapositivas

Campus: Ciudad Universitaria
Facultad: Ingeniería
Materia : Inteligencia Artificial
Semestre: 2022-2
Equipo: 1
Clave: 0406
Participantes:

- Barrera Peña Víctor Miguel
- Espino De Horta Joaquín Gustavo

Profesor: Dr. Ismael Everardo Barcenas Patiño
Título : Proyecto 2
Subtítulo : Fórmula de lógica proposicional valida o no

Comenzamos

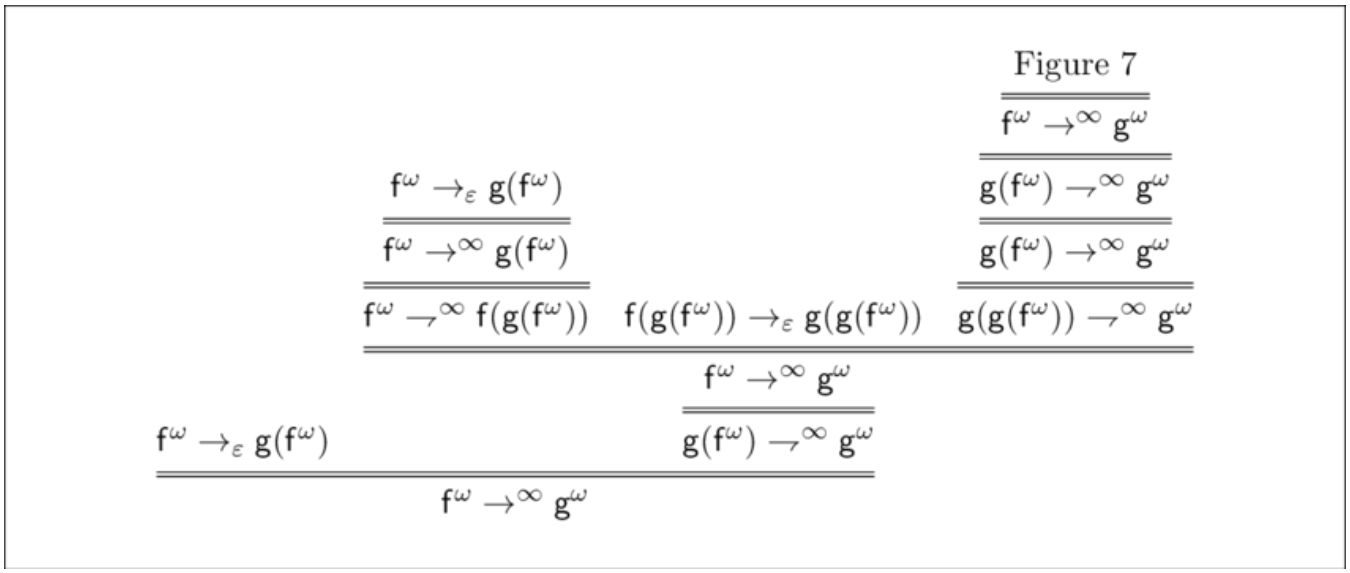
Figure 7
$f^\omega \to_\varepsilon g(f^\omega)$
$f^\omega \to^\infty g^\omega$
$f^\omega \to^\infty g(f^\omega)$
$f^\omega \to^\infty f(g(f^\omega))$
$f(g(f^\omega)) \to_\varepsilon g(g(f^\omega))$
$f^\omega \to^\infty g^\omega$
$g(f^\omega) \to^\infty g^\omega$
$g(f^\omega) \to^\infty g^\omega$
$g(g(f^\omega)) \to^\infty g^\omega$
$f^\omega \to_\varepsilon g(f^\omega)$
$f^\omega \to^\infty g^\omega$
$g(f^\omega) \to^\infty g^\omega$
$f^\omega \to^\infty g^\omega$

Definición del problema

Escribir un programa lógico que determine si una fórmula de lógica proposicional es válida o no.

- Algoritmo basado en el cálculo de secuentes.
- Entrada: una expresión de lógica proposicional.
- Salida: una prueba (en código latex), si existe, de lo contrario, no válida.

Capítulo 0 Estructura del repositorio

El repositorio documenta todo el proceso hecho para realizar el programa , incluyendo los pdf´s

Bibliotecas usadas

`\usespackage{prof}`
`\usespackage{bussproofs}`
Para poder ejecutar el programa es necesario tener instalado un interprete de `prolog` para poder ejecutar el programa.
Para dar la apariencia a las etiquetas de los `prooftree` pues se uso
`\mathcal{I}` `\mathcal{D}`

Capítulo 1 Introducción

"Aquel modo de razonamiento que examina el estado de las cosas afirmadas en las premisas, forma un diagrama de este estado de cosas, percibe en las partes de ese diagrama relaciones no mencionadas expl´ıcitamente en las premisas, se satisface a s´ı mismo mediante experimentos mentales sobre el diagrama, de que estas relaciones siempre subsistir´an, y concluye su verdad necesaria o probable" Charles Sanders Peirce.

Razonar partiendo de premisas mediante reglas para llegar a conclusiones es la tarea que tomamos desde la matemática para poder demostrar, este es uno de los principales objetivos de la lógica. Formalmente se demuestra la validez de un razonamiento mediante el método de deducción.

El método de la deducción se hace usando las leyes de lógica de primer orden o mejor conocidas como **deducción natural**. Para logar obtener un resultado se hace uso de su principal herramienta que es el **cálculo de secuentes**.

Capítulo 2 Desarrollo

Idea de desarrollo del programa

Casos de prueba

Triviales (1 caso)

$$\frac{}{A \vdash A}\ \mathcal{JD}$$

Fáciles (3 casos)

Caso 1

$$\dfrac{\dfrac{\dfrac{}{\vdash \neg A, A}\,\mathcal{JD}}{\vdash \neg A, A}\,\mathcal{D}\neg}{\neg\neg A \vdash A}\,\mathcal{J}\neg$$

Fáciles (3 casos)

Caso 1

$$\dfrac{\dfrac{\dfrac{}{\vdash \neg A, A}\,\mathcal{ID}}{\vdash \neg A, A}\,\mathcal{D}\neg}{\neg\neg A \vdash A}\,\mathcal{I}\neg$$

Caso 2

$$\dfrac{\dfrac{}{A \vdash A}\,\mathcal{ID} \qquad \dfrac{}{B \vdash B}\,\mathcal{ID}}{A, B \vdash A \wedge B}\,\mathsf{R}$$

Caso 3

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,\mathcal{ID}}{\vdash \neg A, A}\,\mathcal{R}\neg}{\vdash A, \neg A}\,\mathcal{RX}}{\vdash A \vee \neg A}\,\mathcal{ANT}$$

Media (3 casos)

Caso 1

$$\dfrac{\dfrac{\dfrac{B \vdash A, A \qquad B, B, \vdash A}{B, A \to B \vdash A}\,\mathcal{I}\to}{A \to B \vdash B \to A}\,\mathcal{D}\to \qquad \dfrac{\dfrac{\dfrac{A \vdash B, B \qquad A, A \vdash B}{A, B \to A \vdash B}\,\mathcal{I}\to}{B \to A \vdash A \to B}\,\mathcal{D}\to}{\vdash (A \vee B) \to (A \leftrightarrow B)}\,D\leftrightarrow$$

Caso 2

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}\,\mathcal{I}\mathcal{D} \qquad A \vdash B}{A \vdash (A \wedge B)}\,\mathcal{D}\wedge \qquad \dfrac{\dfrac{\dfrac{}{A \vdash A}\,\mathcal{I}\mathcal{D}}{A, B \vdash A}\,\mathcal{I} - Debilitamiento}{(A \wedge B) \vdash A}\,\mathcal{I}\wedge}{\vdash A \leftrightarrow (A \wedge B)}\,\mathcal{D}\leftrightarrow$$

Caso 3

$$\dfrac{\dfrac{\dfrac{\vdash A, B}{\neg A \vdash B}\,\mathcal{J}\neg}{\vdash (A \vee \neg B)}\,\mathcal{D}\vee \qquad \dfrac{\dfrac{A, B \vdash}{A \vdash \neg B}\,\mathcal{D}\neg}{\vdash (A \vee \neg B)}\,\mathcal{D}\vee}{\vdash (\neg A \vee B) \wedge (A \vee \neg B)}\,\mathcal{D}\quad Conjugacion$$

Difíciles (3 casos)

Caso 1

(revisar)

$$\dfrac{\dfrac{\dfrac{\vdash A, B}{\neg A \vdash B}\,\mathcal{J}\neg}{\vdash (A \vee \neg B)}\,\mathcal{D}\vee \qquad \dfrac{\dfrac{A \vdash A}{A, A \to B \vdash B}\,\mathcal{D}\neg}{A \to \vdash A \to B}\,\mathcal{D}\vee}{\vdash (A \vee B) \to (A \leftrightarrow B)}\,\mathcal{D}\to$$

Caso 2

Caso 3

Sin solución (1 caso)

Caso 1

$$\frac{}{A \vdash B}\ \mathcal{JD}$$

Código

```
%Definir operaciones//
:-op(1,fx,neg).
:-op(2,xfy,or).
:-op(2,xfy,and).
:-op(2,xfy,imp).
:-op(2,xfy,dimp).

%Win Condition
sq(G,D):-
sq([F],[F]).

%Debilitamiento
sq([F|G],D):- atom(F),
sq(G,D).

sq(G,[F|D]):- atom(F),
sq(G,D).
```

parte 2

```
sq(G,[neg F|D]):-
sq([F|G],D).

%Disyuncion
sq([F or R|G],D):-
sq([F|G],D),
sq([R|G],D).

sq(G,[F or R|D]):-
append([F,R],D,U),
sq(G,U).

%Conjuncion
sq([F and R|G],D):-
append([F,R],G,U),
sq(U,D).
```

Parte 3

```
sq(G,[F and R|D]):-
sq(G,[F|D]),
sq(G,[R|D]).


%Implicacion
sq([F imp R|G],D):-
sq(G,[F|D]),
sq([R|G],D).


sq(G,[F imp R|D]):-
sq([F|G],[R|D]).
```

Explicación código

```
%Doble Implicacion
sq([F dimp R|G],D):-
append([F,R],D,U),
sq(G,U),
append([F,R],G,U),
sq(U,D).


sq(G,[F dimp R|D]):-
sq([F|G],[R|D]),
sq([R|G],[F|D]).
```

Capítulo 3 Conclusión

Barrera Peña Víctor Miguel

Crear un programa en un nuevo paradigma de programación es un reto y más si aquel lenguaje haz hecho pocos ejercicios, además de entrar en conflicto con latex para crear las ecuaciones constituye un reto de buen nivel. Por anteriores motivos puedo decir que si bien hasta el momento no se ha logrado obtener todo a la perfección nos hemos acercado al objetivo de crear un programa que pueda crear los secuentes usando programación lógica.
En lo que respecta, posiblemente con una semana más y un poco de práctica para encontrar el error de prolog es suficiente para poder concluir el proyecto con exito.

Espino de Horta Joaquín Gustavo

El mayor desafío de este proyecto sin duda era el conocer y manipular correctamente el lenguaje argumentativo de Prolog un programa cuya única función es la de comprobar hechos y operaciones lógicas. Si bien el programa a realizar es en cierto punto sencillo, no se podía realizar de la manera correcta por muchos factores de falta de información, donde no se encuentra mucho soporte o documentación adecuada así como ciertas funciones que nos acortaban trabajo.
Apuesto que si nos hubiesen dejado este mismo trabajo, con la gran libertad de hacerlo con un lenguaje de programación de libre elección, se habría llegado a un producto satisfactorio en menos de una semana como si ocurrió con el proyecto anterior. En cuanto al objetivo, cumple en el ámbito teórico así como su propuesta de ser posible a realizar en las propiedades de Prolog.

Anexos para investigación

Charles Sanders Peirce

- La ciencia de la semiótica,http://mastor.cl/blog/wp-content/uploads/2015/08/PEIRCE-CH.-S.-La-Ciencia-de-La-Semi%C3%B3tica.pdf

Fuentes

- FUENTES GUZMAN, D. A. N. I. E. L. C. A. M. I. L. O. (2014). CÁLCULO DE SECUENTES Y GRAFICOS EXISTENCIALES ALFA: DOS ESTRUCTURAS EQUIVALENTES PARA LA LOGICA PROPOSICIONAL. UNIVERSIDAD DEL TOLIMA. Recuperado 7 de abril de 2022, de http://repository.ut.edu.co/bitstream/001/1172/1/RIUT-ABA-spa-2014-C%C3%A1lculo%20de%20secuentes%20y%20gr%C3%A1ficos%2

3. código

```
//Definir operaciones//

:-op(1,fx,neg).
:-op(2,xfy,or).
:-op(2,xfy,and).
:-op(2,xfy,imp).
:-op(2,xfy,dimp).

//Win Condition
sq(G,D):-
sq([F],[F]).

//Debilitamiento
sq([F|G],D):- atom(F),
sq(G,D).

sq(G,[F|D]):- atom(F),
sq(G,D).

//Negacion
sq([neg F|G],D):-
sq(G,[F|D]),
//write(Archivo,"\neg"+G).

sq(G,[neg F|D]):-
sq([F|G],D).

//Disyuncion
sq([F or R|G],D):-
sq([F|G],D),
sq([R|G],D).

sq(G,[F or R|D]):-
append([F,R],D,U),
sq(G,U).

//Conjuncion
sq([F and R|G],D):-
append([F,R],G,U),
sq(U,D).

sq(G,[F and R|D]):-
sq(G,[F|D]),
sq(G,[R|D]).

//Implicacion
sq([F imp R|G],D):-
```

```
sq(G,[F|D]),
sq([R|G],D).

sq(G,[F imp R|D]):-
sq([F|G],[R|D]).

//Doble Implicacion
sq([F dimp R|G],D):-
append([F,R],D,U),
sq(G,U),
append([F,R],G,U),
sq(U,D).

sq(G,[F dimp R|D]):-
sq([F|G],[R|D]),
sq([R|G],[F|D]).
```