Tableaux para chequear consistencia de una base de conocimiento en extensiones de $\mathcal{ALC}$:

Paula Severi

University of Leicester

Curso Noviembre-Diciembre 2018. Facultad de Ingeniería.
Universidad de la República, Montevideo, Uruguay.

Resumen

Algoritmo de Tableaux para chequear consistencia de una base de conocimiento en extensiones de $\mathcal{ALC}$

1. restricciones de cardinalidad
2. roles inversos
3. roles transitivos
4. jerarquía de roles

$\mathcal{ALCQ} = \mathcal{ALC}$ + restricciones de cardinalidad

Sea $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ una interpretación de $\mathcal{ALC}$.

Semántica de $\geq n\ R$ donde n un natural no negativo

$$(\geq n\ R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \mathsf{card}(\mathsf{imagen}(x, C)) \geq n\}$$

Semántica de $\leq n\ R$ donde n un natural no negativo

$$(\leq n\ R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \mathsf{card}(\mathsf{imagen}(x, C)) \leq n\}$$

Notación:

$\mathsf{card}(X)$ es la cardinalidad de conjunto X

$\mathsf{imagen}(x, C) = \{y \in C^{\mathcal{I}} \mid (x, y) \in R^{\mathcal{I}}\}$

Forma Normal de Negación

Dualidad

$$
\begin{array}{rcl}
\neg(\leq n\ R.C) & = & \geq (n+1)R.C \\
\neg(\geq (n+1)\ R.C) & = & \leq nR.C
\end{array}
$$

Usando la dualidad anterior, podemos convertir un concepto a su forma normal de negación.

¿ Cuando paramos de aplicar las reglas?

Necesitamos agregar las relaciones de igualdad $=$ y diferencia $\neq$ para poder tratar las restricciones de cardinalidad.

Tableaux con contradicción

$\mathcal{L}$ tiene una contradicción cuando

- existen a y C tal que $\{C, \neg C\} \subseteq \mathcal{L}(a)$ o cuando
- existe un nodo x tal que $\leq nR.C \in \mathcal{L}(x)$, $R \in \mathcal{L}(x, y_i)$, $C \in \mathcal{L}(y_i)$, $y_i \neq y_j$ para todo $i, j \in \{1, \ldots n+1\}$.

Bloqueo

Un nodo cuya etiqueta es x se dice que está bloqueado por un nodo con etiqueta y si no existe y tal que $\mathcal{L}(x, y) \neq \emptyset$ o se cumple lo siguiente:

1. x es una variable (no es un individuo),
2. y es un ancestro de x,
3. $\mathcal{L}(x) \subseteq \mathcal{L}(y)$.

Reglas de expansión para $\mathcal{ALCQ}$ (primera parte)

Regla $\sqcap$: Si $C \sqcap D \in \mathcal{L}(x)$ y $\{C, D\} \not\subseteq \mathcal{L}(x)$ entonces ponemos $\mathcal{L}(x) \leftarrow \{C, D\}$.

Regla $\sqcup$: Si $C \sqcup D \in \mathcal{L}(x)$ y $\{C, D\} \cap \mathcal{L}(x) = \emptyset$ entonces ponemos $\mathcal{L}(x) \leftarrow \{C\}$ o $\mathcal{L}(x) \leftarrow \{D\}$.

Regla $\exists$: Si x no está bloqueado, $\exists R.C \in \mathcal{L}(x)$ y no existe y tal que $R \in \mathcal{L}(x, y)$ y $C \in \mathcal{L}(y)$ entonces

1. creamos un nodo nuevo y
2. ponemos $\mathcal{L}(x, y) = \{R\}$,
3. ponemos $\mathcal{L}(y) = \{C\}$.

Regla $\forall$: Si $\forall R.C \in \mathcal{L}(x)$ y
existe y tal que $R \in \mathcal{L}(x, y)$ y $C \notin \mathcal{L}(y)$ entonces
ponemos $\mathcal{L}(y) \leftarrow C$

Regla $\mathcal{T}$: Si $C \in \mathcal{T}$ y $C \notin \mathcal{L}(x)$ entonces $\mathcal{L}(x) \leftarrow C$.

Regla $\geq$: si $\geq nR.C \in \mathcal{L}(x)$, x no está bloqueado y no hay $y_1, \ldots, y_n$ tal que $R \in \mathcal{L}(x, y_i)$, $C \in \mathcal{L}(y_i)$, $y_i \neq y_j$ para $i, j \in \{1, \ldots n\}$ entonces

1. creamos nodos nuevos $y_1, \ldots y_n$.
2. ponemos $\mathcal{L}(x, y_i) = \{R\}$, $\mathcal{L}(y_i) = \{C\}$ y $y_i \neq y_j$ para $i, j \in \{1, \ldots n\}$.

Regla selección: si $\leq nR.C \in \mathcal{L}(x)$ y existe y tal que $R(x, y)$, $C \notin \mathcal{L}(y)$, $\mathsf{FNN}(\neg C) \notin \mathcal{L}(y)$ entonces

ponemos $\mathcal{L}(y) \leftarrow C$ o $\mathcal{L}(y) \leftarrow \mathsf{FNN}(\neg C)$.

Regla $\leq$: si $\leq nR.C \in \mathcal{L}(x)$, existen $y_1, \ldots, y_{n+1}$ tal que $R \in \mathcal{L}(x, y_i)$, $C \in \mathcal{L}(y_i)$ para $i, j \in \{1, \ldots n + 1\}$ y existe $i, j \in \{1, \ldots n\}$ tal que $y_i \neq y_j$ no se cumple entonces

1. ponemos $\mathcal{L}(y_i) \leftarrow \mathcal{L}(y_j)$,
2. ponemos $\mathcal{L}(y_i, z) \leftarrow \mathcal{L}(y_j, z)$ y $\mathcal{L}(z, y_i) \leftarrow \mathcal{L}(z, y_j)$,
3. ponemos $y_i = y_j$ y $\mathcal{L}(y_j) = \mathcal{L}(y_j, z) = \mathcal{L}(z, y_j) = \emptyset$.

Ejemplo: uso de la segunda condición de contradicción

Probamos inconsistencia de $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ usando Tableaux.

$$
\begin{aligned}
\mathcal{T} = \; & \{\mathtt{Hombre} \sqsubseteq \geq 3\mathtt{tieneHijo}.\mathtt{Hombre} \\
& \mathtt{Hombre} \sqsubseteq \leq 1\mathtt{tieneHijo}.\mathtt{Hombre}\} \\
\mathcal{A} = \; & \{\mathtt{Hombre}(\mathtt{mario})\}
\end{aligned}
$$

Ejemplo del uso de la regla $\leq$

Sea $\mathcal{K} = (\emptyset, \mathcal{A})$.

$$
\begin{aligned}
\mathcal{A} = \ & \{\texttt{tieneHijo}(\texttt{mario}, \texttt{luis}),\\
& \texttt{tieneHijo}(\texttt{mario}, \texttt{jorge}),\\
& \texttt{Hombre}(\texttt{luis}),\\
& \texttt{Hombre}(\texttt{jorge})\\
& \leq 2\texttt{tieneHijo}.\top(\texttt{mario})\}
\end{aligned}
$$

Demostramos que

$$\mathcal{K} \not\models \forall\texttt{tieneHijo}.\texttt{Hombre}(\texttt{mario})$$

usando el algoritmo de tableaux.

Ejemplo: uso de la regla de selección

Probamos consistencia de $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ usando Tableaux.

$$\begin{aligned} \mathcal{T} &= \{\top \sqsubseteq \leq 2\mathtt{t}.\mathtt{H}\} \\ \mathcal{A} &= \{\geq 3\mathtt{t}.\top(\mathtt{j})\} \end{aligned}$$

Ejemplo: uso de la regla de selección

Probamos inconsistencia de $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ usando Tableaux.

$$\begin{aligned}
\mathcal{T} = \quad & \{\mathtt{Hombre} \sqsubseteq \geq 3\mathtt{tieneHijo.Hombre} \\
& \mathtt{Hombre} \sqsubseteq \leq 1\mathtt{tieneHijo.Informatico} \\
& \mathtt{Hombre} \sqsubseteq \leq 1\mathtt{tieneHijo.}\neg\mathtt{Informatico}\} \\
\mathcal{A} = \quad & \{\mathtt{Hombre(Mario)}\}
\end{aligned}$$

$\mathcal{SI} = \mathcal{ALC}$ + roles transitivos e inversos

R^- es el inverso de R. Identificamos R^{--} con R^-. El conjunto de roles es $\mathsf{N_R} \cup \{R^- \mid R \in \mathsf{N_R}\}$.

$$C, D := \bot \mid \top \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$$

donde $A \in \mathsf{N_C}$ y $R \in \mathsf{N_R} \cup \{R^- \mid R \in \mathsf{N_R}\}$.

Ejemplo con roles transitivos

Sea $C = A \sqcap \exists R.A \sqcap \forall R.\exists R.A$, $a : C$ y R transitivo.

$$\begin{array}{ll} a & \mathcal{L}(a) = \{C, A, \exists R.A, \forall R.\exists R.A\} \\ \Big\downarrow R & \\ x & \mathcal{L}(x) = \{A, \exists R.A, \forall R.\exists R.A\} \end{array}$$

Como R es transitivo, $\forall R.\exists R.A$ se propaga a lo largo de los arcos etiquetados con R.

No podemos asegurar terminación porque ahora agregamos un concepto que tiene el mismo tamaño que el concepto descompuesto.

Para asegurar la terminación del algoritmo agregamos el concepto de bloqueo indirecto. Las reglas no generadoras (de nodos nuevos) se pueden aplicar si el nodo no está indirectamente bloqueado.

Bloqueo para roles transitivos e inversos

Nodo bloqueado

Un nodo x está bloqueado por un ancestro y si x es una variable (no es un individuo) y se cumple una de estas clausulas:

- $\mathcal{L}(x) = \mathcal{L}(y)$ o
- algún ancestro de x está bloqueado.

Si se cumple la segunda entonces x está indirectamente bloqueado, si no se cumple entonces se dice que x está directamente bloqueado.

Vecino

Un nodo es R-vecino de y si $R \in \mathcal{L}(x, y)$ o $R^{-} \in \mathcal{L}(y, x)$.

Reglas de expansión para $\mathcal{SI}$

Regla $\sqcap$: Si x no está indirectamente bloqueado, $C \sqcap D \in \mathcal{L}(x)$ y $\{C, D\} \nsubseteq \mathcal{L}(x)$ entonces ponemos $\mathcal{L}(x) \leftarrow \{C, D\}$.

Regla $\sqcup$: Si x no está indirectamente bloqueado, $C \sqcup D \in \mathcal{L}(x)$ y $\{C, D\} \cap \mathcal{L}(x) = \emptyset$ entonces ponemos $\mathcal{L}(x) \leftarrow \{C\}$ o $\mathcal{L}(x) \leftarrow \{D\}$.

Regla $\exists$: Si x no está bloqueado, $\exists R.C \in \mathcal{L}(x)$ y no existe ningún y que sea R-vecino de x y que $C \in \mathcal{L}(y)$ entonces

1. creamos un nodo nuevo y
2. ponemos $\mathcal{L}(x, y) = \{R\}$,
3. ponemos $\mathcal{L}(y) = \{C\}$.

Regla $\forall$: Si x no está indirectamente bloqueado, $\forall R.C \in \mathcal{L}(x)$ y existe y R-vecino de x tal que $C \notin \mathcal{L}(y)$ entonces ponemos $\mathcal{L}(y) \leftarrow C$

Regla $\forall_+$: Si x no está indirectamente bloqueado, $\forall R.C \in \mathcal{L}(x)$, R o el inverso de R es transitivo, hay un vecino y de x tal que $\forall R.C \notin \mathcal{L}(y)$ entonces ponemos $\mathcal{L}(y) \leftarrow \forall R.C$

Regla $\mathcal{T}$: Si x no está indirectamente bloqueado, $C \in \mathcal{T}$ y $C \notin \mathcal{L}(x)$ entonces $\mathcal{L}(x) \leftarrow C$.

Ejemplo

Sea $\mathcal{K}$ la siguiente base de conocimiento:

$$\texttt{hasChild}.\texttt{Human}(\texttt{john})$$
$$\texttt{Human} \sqsubseteq \forall \texttt{hasParent}.\texttt{Human}$$

`hasChild` es la inversa de `hasParent`. Probamos que

$$\mathcal{K} \models \texttt{Human}(\texttt{john})$$

Bloqueo Dinámico es esencial para $\mathcal{SI}$

Sea P transitivo, $a : D$, $C = \forall R^{-}.\forall P^{-}.\forall P^{-}.\forall S^{-}.\neg A$,
$E = (\exists R.\top \sqcap \exists P.\top \sqcap \forall R.C \sqcap \forall P.E)$, $D = A \sqcap \exists S.E$.

$$\begin{tikzcd} & a \arrow[d, "S"] & \\ & x \arrow[dl, "R"'] \arrow[dr, "P"] & \\ y & & z \end{tikzcd}$$

$$\mathcal{L}(a) = \{D, A, \exists S.E\}$$

$$\mathcal{L}(x) = \{E, \exists R.\top, \exists P.\top, \forall R.C, \forall P.E\}$$

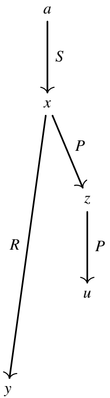
$$\mathcal{L}(z) = \mathcal{L}(x)$$

$$\mathcal{L}(y) = \{C\}$$

Está bloqueado pero todavía no detectamos la inconsistencia.

Bloqueo Dinámico es esencial para $\mathcal{SI}$

Sea P transitivo, $a : D$, $C = \forall R^{-}.\forall P^{-}.\forall P^{-}.\forall S^{-}.\neg A$,
$E = (\exists R.\top \sqcap \exists P.\top \sqcap \forall R.C \sqcap \forall P.E)$, $D = A \sqcap \exists S.E$.



$\mathcal{L}(a) = \{D, A, \exists S.E, \neg A\}$

$\mathcal{L}(x) = \{E, \exists R.\top, \exists P.\top, \forall R.C, \forall P.E, \forall P^{-}.\forall P^{-}.\forall S^{-}.\neg A, \forall S^{-}.\neg A\}$

$\mathcal{L}(z) = \{E, \exists R.\top, \exists P.\top, \forall R.C, \forall P.E, \forall P^{-}.\forall P^{-}.\forall S^{-}.\neg A, \forall P^{-}.\forall S^{-}.\neg A\}$

$\mathcal{L}(u) = \{\forall P^{-}.\forall P^{-}.\forall S^{-}.\neg A\}$

$\mathcal{L}(y) = \{C, \forall R^{-}.\forall P^{-}.\forall P^{-}.\forall S^{-}.\neg A\}$

$\mathcal{SHIQ} = \mathcal{ALCQ}+$ roles transitivos, inversos, jerarquía de roles

Base de conocimientos $\mathcal{K} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ donde $\mathcal{R}$ es un conjunto de $R \sqsubseteq S$ y $\mathsf{Trans}(R)$ donde R es un nombre de rol o el inverso de un nombre de rol.

subrol

R es un subrol de S si R y S estan relacionados por medio de la clausura reflexo-transitiva de $\sqsubseteq$ en $\mathcal{R}$.

Sucesor

y es un S-sucesor de x si $R \in \mathcal{L}(x, y)$, R es un subrol de S.

$\mathcal{SHIQ}$

- $\mathcal{SHIQ}$ no satisface la propiedad de la finitud del modelo (existen bases que solo admiten modelos infinitos).
- El bloqueo es más complicado: *bloqueo en parejas*.

Ejemplo donde se necesita el bloqueo de a parejas

$$A \equiv \neg C \sqcap (\leqslant 1\ F) \sqcap \exists F^{-}.B \sqcap \forall R^{-}.\exists F^{-}.B \qquad F \sqsubseteq R$$
$$B \equiv C \sqcap (\leqslant 1\ F) \sqcap \exists F.\neg C \qquad A(a)$$

R es transitivo.

$$\begin{array}{ll}
a & \mathcal{L}(a) = \{A, \neg C, (\leqslant 1\ F), \exists F^{-}.B, \forall R^{-}.\exists F^{-}.B\} \\
\downarrow F^{-} & \\
x_1 & \mathcal{L}(x_1) = \{B, \exists F^{-}.B, \forall R^{-}.\exists F^{-}.B, C, \leqslant 1\ F, \exists F.\neg C\} \\
\downarrow F^{-} & \\
x_2 & \mathcal{L}(x_2) = \{B, \exists F^{-}.B, \forall R^{-}.\exists F^{-}.B, C, \leqslant 1\ F, \exists F.\neg C\}
\end{array}$$

Sin bloqueo a parejas, x_2 estaria bloqueado y no detectamos la inconsistencia.

Ejemplo (cont)

Mostrar como da la inconsistencia

$$A \equiv \neg C \sqcap (\leqslant 1\ F) \sqcap \exists F^{-}.B \sqcap \forall R^{-}.\exists F^{-}.B \qquad F \sqsubseteq R$$
$$B \equiv C \sqcap (\leqslant 1\ F) \sqcap \exists F.\neg C \qquad A(a)$$

R es transitivo.

$$\begin{array}{ll} a & \mathcal{L}(a) = \{A, \neg C, (\leqslant 1\ F), \exists F^{-}.B, \forall R^{-}.\exists F^{-}.B\} \\ \downarrow F^{-} & \\ x_1 & \mathcal{L}(x_1) = \{B, \exists F^{-}.B, \forall R^{-}.\exists F^{-}.B, C, \leqslant 1\ F, \exists F.\neg C\} \\ \downarrow F^{-} & \\ x_2 & \mathcal{L}(x_2) = \{B, \exists F^{-}.B, \forall R^{-}.\exists F^{-}.B, C, \leqslant 1\ F, \exists F.\neg C\} \end{array}$$

Completar!

Bloqueo para $\mathcal{SHIQ}$

Nodo bloqueado

Un nodo x está bloqueado si es una variable (no es un individuo) y se cumple una de estas clausulas:

- existen ancestros x', y, y' de x tal que

$$y' \longrightarrow y \longrightarrow \ldots \longrightarrow x' \longrightarrow x$$

  - y es una variable,
  - x es un sucesor de x', y es un sucesor de y'
  - $\mathcal{L}(x) = \mathcal{L}(y)$, $\mathcal{L}(x') = \mathcal{L}(y')$
  - $\mathcal{L}(x', x) = \mathcal{L}(y', y)$
- algún ancestro de x está bloqueado.
- no hay ningún nodo y tal que $\mathcal{L}(y, x) \neq \emptyset$.

Si se cumple la segunda clausula entonces x está indirectamente bloqueado, si no se cumple entonces se dice que x está directamente bloqueado.

Ejemplo en $\mathcal{SHIQ}$ que solo admite modelos infinitos

$$\boxed{A(a) \qquad \leqslant 0\, R^{-1}(a) \qquad A \sqsubseteq \leqslant 1R^{-1} \sqcap \forall R.A \sqcap \exists R.A}$$

$$a \xrightarrow[R]{} x_1 \xrightarrow[R]{} x_2 \xrightarrow[R]{} x_3 \xrightarrow[R]{} \ldots$$

$$\mathcal{L}(a) = \{A, \leqslant 0R^{-1}\}$$
$$\mathcal{L}(x_1) = \{A, \leqslant 1R^{-1} \sqcap \forall R.A \sqcap \exists R.A, \leqslant 1R^{-1}, \forall R.A, \exists R.A\}$$
$$\mathcal{L}(x_2) = \{A, \leqslant 1R^{-1} \sqcap \forall R.A \sqcap \exists R.A, \leqslant 1R^{-1}, \forall R.A, \exists R.A\}$$
$$\vdots$$

Modelo canónico en $\mathcal{SHIQ}$

$A(a)$	$\leqslant 0\ R^{-1}(a)$	$A \sqsubseteq \leqslant 1 R^{-1} \sqcap \forall R.A \sqcap \exists R.A$

$$a \xrightarrow[R]{} x_1 \xrightarrow[R]{} x_2 \xrightarrow[R]{} x_3$$

$$\mathcal{L}(a) = \{A, \leqslant 0\ R^{-1}\} \qquad \mathcal{L}(x_1) = \mathcal{L}(x_2) = \mathcal{L}(x_3) = \{A, \leqslant 1 R^{-1} \sqcap \forall R.A \sqcap \exists R.A, \leqslant 1 R^{-1}, \forall R.A, \exists R.A\}$$

Con el bloqueo, el grafo termina y es finito. El dominio del modelo no son los nodos del grafo (estos son finitos).

El modelo canónico se construye con los caminos

$$\Delta^{\mathcal{I}} = \{[a], [a, x_1], [a, x_1, x_2], [a, x_1, x_2], [a, x_1, x_2, x_2], \ldots\}$$

Otro ejemplo que solo admite modelos infinitos

Sea $\mathcal{K}$ la siguiente base de conocimiento.

$$\begin{array}{l} \mathsf{Trans}(R) \quad F \sqsubseteq R \\ \neg C \sqcap \exists F^{-}.(C \sqcap \leq 1 F) \sqcap \forall R^{-}.(\exists F^{-}.(C \sqcap \leq 1 F)) \end{array}$$

Todos los modelos de $\mathcal{K}$ tienen un dominio infinito.

Ejemplo con roles transitivos y restricciones de cardinalidad

Probamos que la siguiente base de conocimiento es consistente:

$$\mathtt{Humano} \sqsubseteq \exists \mathtt{tienePadre}.\top$$
$$\mathtt{tienePadre} \sqsubseteq \mathtt{tieneAncestro}$$
$$\forall \mathtt{tieneAncestro}.\mathtt{Humano}(\mathtt{juan})$$
$$\geq 2\mathtt{tienePadre}.\top(\mathtt{juan})$$
$$\mathtt{tieneAncestro} \circ \mathtt{tieneAncestro} \sqsubseteq \mathtt{tieneAncestro}$$

Ejemplo donde se necesita el bloqueo en parejas

Probamos que la siguiente base de conocimientos es inconsistente.

$$F \sqsubseteq R \quad R \circ R \sqsubseteq R$$
$$(\neg C \sqcap (\leq 1 F) \sqcap \exists F^{-}.D \sqcap \forall R^{-}.\exists F^{-}.D)(a)$$

donde D es una abreviación para $C \sqcap (\leq 1 F) \sqcap \exists F.\neg C$.

Complejidad del Algoritmo de Tableaux

1. Complejidad de este algoritmo de Tableau es alta : N2Exp para $\mathcal{SROIQ}$ (peor caso) y NExp para $\mathcal{SHIQ}$ (peor caso).

2. Complejidad (del problema) de la consistencia de $\mathcal{SROIQ}$ es NEXP y en $\mathcal{SHIQ}$ es EXP, i.e. *hay algoritmos mejores que los de Tableau*

Complejidad del Algoritmo de Tableaux

1. Complejidad de este algoritmo de Tableau es alta : N2Exp para $\mathcal{SROIQ}$ (peor caso) y NExp para $\mathcal{SHIQ}$ (peor caso).

2. Complejidad (del problema) de la consistencia de $\mathcal{SROIQ}$ es NEXP y en $\mathcal{SHIQ}$ es EXP, i.e. *hay algoritmos mejores que los de Tableau*

3. Todos los razonadores standard usan Tableaux: Fact, Pellet, Hermit, etc. Porqué

Complejidad del Algoritmo de Tableaux

1. Complejidad de este algoritmo de Tableau es alta : N2Exp para $\mathcal{SROIQ}$ (peor caso) y NExp para $\mathcal{SHIQ}$ (peor caso).
2. Complejidad (del problema) de la consistencia de $\mathcal{SROIQ}$ es NEXP y en $\mathcal{SHIQ}$ es EXP, i.e. *hay algoritmos mejores que los de Tableau*
3. Todos los razonadores standard usan Tableaux: Fact, Pellet, Hermit, etc. Porqué el peor caso no sucede en la práctica+ OPTIMIZACIONES!!!

Bibliografía para esta clase

- Capítulo 5 de *Foundations of Semantic Web Technologies.* Pascal Hitzler, Markus Krötzsch and Sebastian Rudolph. 2009 Chapman & Hall/CRC.
- Ian Horrocks, Ulrike Sattler: A Description Logic with Transitive and Inverse Roles and Role Hierarchies. J. Log. Comput. 9(3): 385-410 (1999)
- Franz Baader, Ulrike Sattler. An overview of tableau algorithms for Description Logics. Studia Logica 2000.