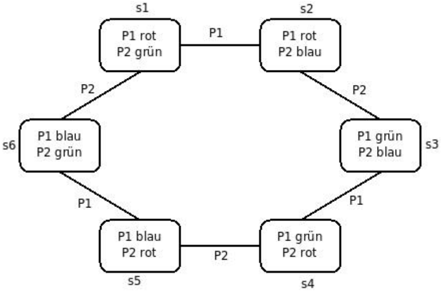
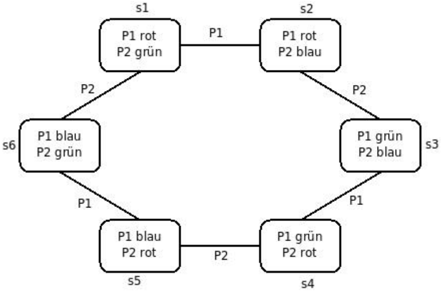
Systeme mit mehreren Agenten und Wissensbasen

Marcus Roseneck

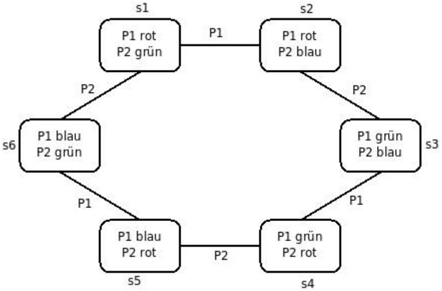
4. Juni 2008



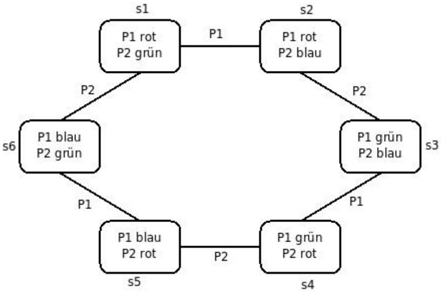
- $\mathcal{M} = (\mathcal{S}, \pi, K_1, ..., K_n)$



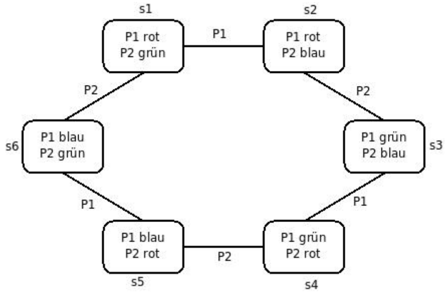
- $\mathcal{M} = (\mathcal{S}, \pi, K_1, ..., K_n)$
- $\mathcal{S} = (s_1, s_2, s_3, s_4, s_5, s_6)$



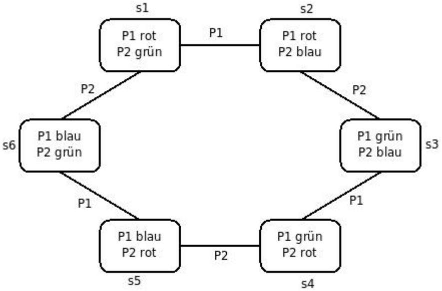
- $\mathcal{M} = (\mathcal{S}, \pi, K_1, ..., K_n)$
- $\mathcal{S} = (s_1, s_2, s_3, s_4, s_5, s_6)$
- $\Phi = \{$P1 rot, P1 blau,P1 grün,P2 rot, P2 blau, P2 grün$\}$



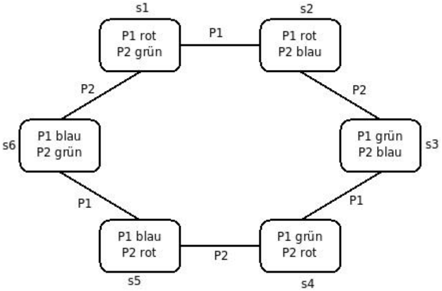
- $\mathcal{M} = (\mathcal{S}, \pi, K_1, ..., K_n)$
- $\mathcal{S} = (s_1, s_2, s_3, s_4, s_5, s_6)$
- $\Phi = \{$P1 rot, P1 blau,P1 grün,P2 rot, P2 blau, P2 grün$\}$
- $\pi(\Phi) \rightarrow \{\mathit{true}, \mathit{false}\}$ für alle $s \in \mathcal{S}, \varphi \in \Phi$



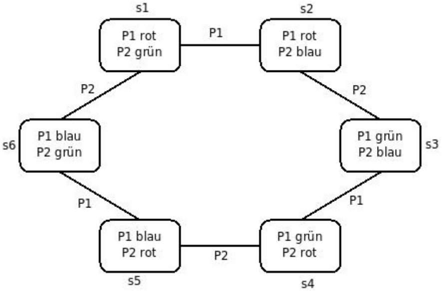
- $\mathcal{M} = (\mathcal{S}, \pi, K_1, \ldots, K_n)$
- $\mathcal{S} = (s_1, s_2, s_3, s_4, s_5, s_6)$
- $\Phi = \{$P1 rot, P1 blau,P1 grün,P2 rot, P2 blau, P2 grün$\}$
- $\pi(\Phi) \rightarrow \{true, false\}$ für alle $s \in \mathcal{S}, \varphi \in \Phi$
- $K1 = \{(s1, s2)(s3, s4), (s5, s6), \ldots\}$ (K_i sind Äquivalenzrelationen)



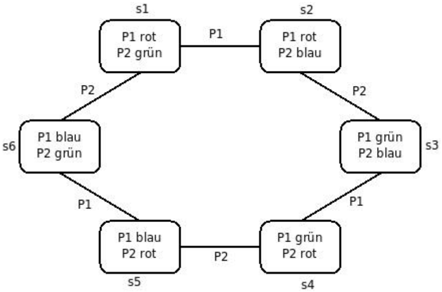
- $\mathcal{M} = (\mathcal{S}, \pi, K_1, ..., K_n)$
- $\mathcal{S} = (s_1, s_2, s_3, s_4, s_5, s_6)$
- $\Phi = \{$P1 rot, P1 blau,P1 grün,P2 rot, P2 blau, P2 grün$\}$
- $\pi(\Phi) \rightarrow \{true, false\}$ für alle $s \in \mathcal{S}, \varphi \in \Phi$
- $K1 = \{(s1, s2)(s3, s4), (s5, s6), ...\}$ (K_i sind Äquivalenzrelationen)
- $K2 = \{(s2, s3), (s4, s5), (s1, s6), ...\}$



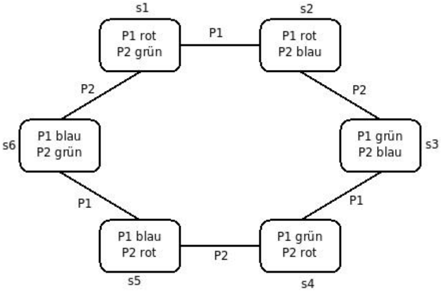
- $(\mathcal{M}, s) \models \varphi \iff \pi(s)(\varphi) = \textbf{true}$



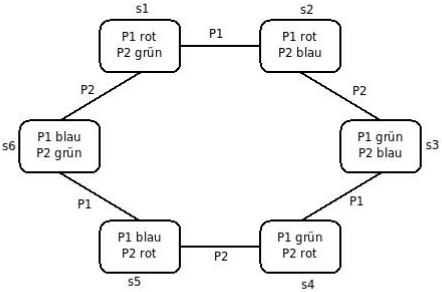
- $(\mathcal{M}, s) \models \varphi \iff \pi(s)(\varphi) = \textbf{true}$
- $(\mathcal{M}, s) \models \varphi \wedge \psi \iff (\mathcal{M}, s) \models \varphi \textit{ and } (\mathcal{M}, s) \models \psi$



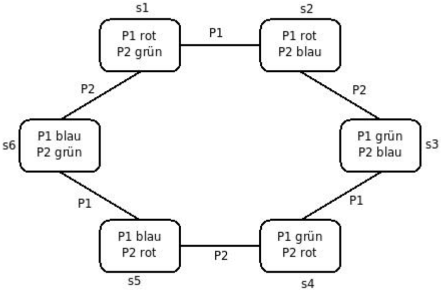
- $(\mathcal{M}, s) \models \varphi \iff \pi(s)(\varphi) = \textbf{true}$
- $(\mathcal{M}, s) \models \varphi \wedge \psi \iff (\mathcal{M}, s) \models \varphi$ *and* $(\mathcal{M}, s) \models \psi$
- $(\mathcal{M}, s) \models K_i\varphi \iff (\mathcal{M}, t) \models \varphi$ für alle φ so dass $(s, t) \in K_i$



- $(\mathcal{M}, s) \models \varphi \iff \pi(s)(\varphi) = \textbf{true}$
- $(\mathcal{M}, s) \models \varphi \wedge \psi \iff (\mathcal{M}, s) \models \varphi$ *and* $(\mathcal{M}, s) \models \psi$
- $(\mathcal{M}, s) \models K_i\varphi \iff (\mathcal{M}, t) \models \varphi$ für alle φ so dass $(s, t) \in K_i$
- $(\mathcal{M}, s) \models C_G\varphi \iff (\mathcal{M}, s) \models E_G^k\varphi$ für $k = 1, 2, \ldots$



- $(\mathcal{M}, s) \models \varphi \Longleftrightarrow \pi(s)(\varphi) = \textbf{true}$
- $(\mathcal{M}, s) \models \varphi \wedge \psi \Longleftrightarrow (\mathcal{M}, s) \models \varphi$ *and* $(\mathcal{M}, s) \models \psi$
- $(\mathcal{M}, s) \models K_i\varphi \Longleftrightarrow (\mathcal{M}, t) \models \varphi$ für alle φ so dass $(s, t) \in K_i$
- $(\mathcal{M}, s) \models C_G\varphi \Longleftrightarrow (\mathcal{M}, s) \models E_G^k\varphi$ für $k = 1, 2, \ldots$
- $(\mathcal{M}, s) \models E_G^k\varphi \Longleftrightarrow (M, t) \models \varphi$ für alle t, die von s aus in k Schritten G-erreichbar sind (G ist die Menge aller Agenten).



- $(\mathcal{M}, s) \models \varphi \iff \pi(s)(\varphi) = \textbf{true}$
- $(\mathcal{M}, s) \models \varphi \wedge \psi \iff (\mathcal{M}, s) \models \varphi$ *and* $(\mathcal{M}, s) \models \psi$
- $(\mathcal{M}, s) \models K_i\varphi \iff (\mathcal{M}, t) \models \varphi$ für alle φ so dass $(s, t) \in K_i$
- $(\mathcal{M}, s) \models C_G\varphi \iff (\mathcal{M}, s) \models E_G^k\varphi$ für $k = 1, 2, \ldots$
- $(\mathcal{M}, s) \models E_G^k\varphi \iff (M, t) \models \varphi$ für alle t, die von s aus in k Schritten G-erreichbar sind (G ist die Menge aller Agenten).
- $(\mathcal{M}, s) \models D_G\varphi \iff (\mathcal{M}, t) \models \varphi$ für alle t mit $(s, t) \in \cap_{i \in G} K_i$

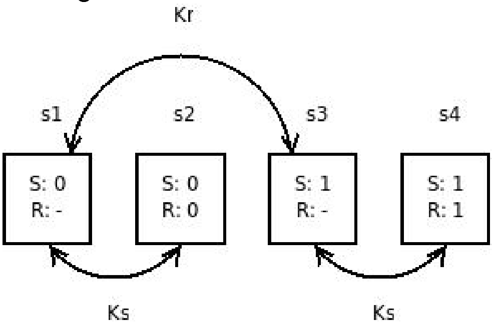
- Ein Sender S sendet Bits (0 oder 1) an einen Empfänger R, der den Erhalt der Nachricht bestätigt (ack message), sobald er sie bekommen hat.

- Ein Sender S sendet Bits (0 oder 1) an einen Empfänger R, der den Erhalt der Nachricht bestätigt (ack message), sobald er sie bekommen hat.
- Leider ist die Leitung fehlerhaft und es kann sein, dass beide Nachrichten verloren gehen.

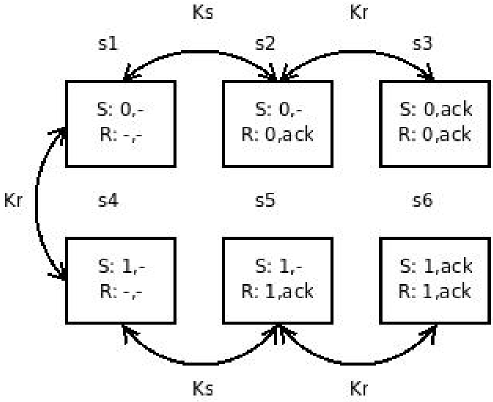
- Ein Sender S sendet Bits (0 oder 1) an einen Empfänger R, der den Erhalt der Nachricht bestätigt (ack message), sobald er sie bekommen hat.
- Leider ist die Leitung fehlerhaft und es kann sein, dass beide Nachrichten verloren gehen.
- Eine Nachricht wird entweder in derselben Runde empfangen, in der sie gesendet wurde, oder sie ist verloren.

- Ein Sender S sendet Bits (0 oder 1) an einen Empfänger R, der den Erhalt der Nachricht bestätigt (ack message), sobald er sie bekommen hat.
- Leider ist die Leitung fehlerhaft und es kann sein, dass beide Nachrichten verloren gehen.
- Eine Nachricht wird entweder in derselben Runde empfangen, in der sie gesendet wurde, oder sie ist verloren.
- Wegen der Übertragungsunsicherheit, sendet S dieselbe Nachricht so lange, bis S eine Empfangsbestätigung von R bekommt und R sendet die ack-Message ebenfalls jede Runde nach dem Erhalt eines Bits.
  Wie lässt sich diese Situation modellieren?

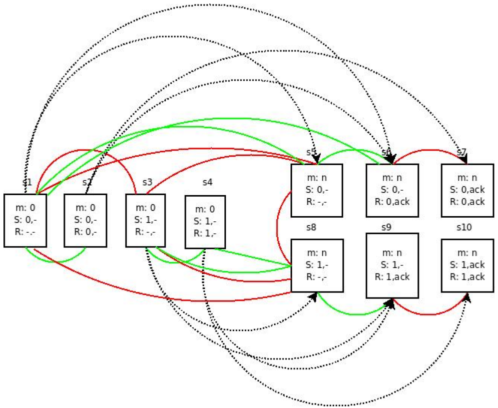
- Zu Beginn sind die Zustände offenbar dieser Art:



- Und in jeder folgenden Runde dieser:



- bestimmte Zustände können zudem in bestimmte andere Zustände übergehen:


s1
m: 0
S: 0,-
R: -,-
s2
m: 0
S: 0,-
R: 0,-
s3
m: 0
S: 1,-
R: -,-
s4
m: 0
S: 1,-
R: 1,-
s5
m: n
S: 0,-
R: -,-
s6
m: n
S: 0,-
R: 0,ack
s7
m: n
S: 0,ack
R: 0,ack
s8
m: n
S: 1,-
R: -,-
s9
m: n
S: 1,-
R: 1,ack
s10
m: n
S: 1,ack
R: 1,ack

- Die betrachteten Zustände hängen von der jeweiligen Runde ab, in der man sich befindet.

- Die betrachteten Zustände hängen von der jeweiligen Runde ab, in der man sich befindet.
- Der Zustand einzelner Agenten unterscheidet sich voneinander und von nicht-agentenspezifischen Zuständen.

- Die betrachteten Zustände hängen von der jeweiligen Runde ab, in der man sich befindet.
- Der Zustand einzelner Agenten unterscheidet sich voneinander und von nicht-agentenspezifischen Zuständen.
- Das Wissen eines Agenten kann sich verändern.

- Ein System lässt sich als eine Menge von Zuständen über eine Menge von Zeitpunkten charakterisieren. Ein Kartenspiel besteht zu einem beliebigen Zeitpunkt aus einer Menge von Spielern sowie Zusatzinformationen, etwa über die nichtverteilten Karten.

- Ein System lässt sich als eine Menge von Zuständen über eine Menge von Zeitpunkten charakterisieren. Ein Kartenspiel besteht zu einem beliebigen Zeitpunkt aus einer Menge von Spielern sowie Zusatzinformationen, etwa über die nichtverteilten Karten.
- Sein *globaler Zustand* lässt sich als n+1-Tupel $(s_e, s_1, ..., s_n)$ für n Agenten beschreiben, wobei s_e der *Umgebungszustand* ist, der alle relevanten Informationen enthält, die keinem einzelnen Agenten zugeschrieben werden können und $(s_1, ..., s_n)$ der *lokale Zustand* der Agenten $1, ..., n$.

- Ein *Durchgang* in einem System ist eine Funktion von den natürlichen Zahlen auf eine Folge von globalen Zuständen eines Systems.

- Ein *Durchgang* in einem System ist eine Funktion von den natürlichen Zahlen auf eine Folge von globalen Zuständen eines Systems.
- L_e :=Menge möglicher Umgebungszustände.

- Ein *Durchgang* in einem System ist eine Funktion von den natürlichen Zahlen auf eine Folge von globalen Zuständen eines Systems.
- L_e :=Menge möglicher Umgebungszustände.
- L_i :=Menge möglicher lokaler Zustände von Agent i.

- Ein *Durchgang* in einem System ist eine Funktion von den natürlichen Zahlen auf eine Folge von globalen Zuständen eines Systems.
- L_e :=Menge möglicher Umgebungszustände.
- L_i :=Menge möglicher lokaler Zustände von Agent i.
- $G := L_e x L_1 x ... x L_n$ Menge der globalen Zustände.

- Ein *Durchgang* in einem System ist eine Funktion von den natürlichen Zahlen auf eine Folge von globalen Zuständen eines Systems.
- L_e :=Menge möglicher Umgebungszustände.
- L_i :=Menge möglicher lokaler Zustände von Agent i.
- $G := L_e x L_1 x ... x L_n$ Menge der globalen Zustände.
- Ein Paar (r, m) heißt 'Punkt' in einem System und bezeichnet Durchgang r zum Zeitpunkt m.

- Ein *Durchgang* in einem System ist eine Funktion von den natürlichen Zahlen auf eine Folge von globalen Zuständen eines Systems.
- L_e :=Menge möglicher Umgebungszustände.
- L_i :=Menge möglicher lokaler Zustände von Agent i.
- $G := L_e x L_1 x ... x L_n$ Menge der globalen Zustände.
- Ein Paar (r, m) heißt 'Punkt' in einem System und bezeichnet Durchgang r zum Zeitpunkt m.
- $r(m) := (s_e, s_1, ..., s_n)$ zum Zeitpunkt m.

- Ein *Durchgang* in einem System ist eine Funktion von den natürlichen Zahlen auf eine Folge von globalen Zuständen eines Systems.
- L_e :=Menge möglicher Umgebungszustände.
- L_i :=Menge möglicher lokaler Zustände von Agent i.
- $G := L_e x L_1 x ... x L_n$ Menge der globalen Zustände.
- Ein Paar (r, m) heißt 'Punkt' in einem System und bezeichnet Durchgang r zum Zeitpunkt m.
- $r(m) := (s_e, s_1, ..., s_n)$ zum Zeitpunkt m.
- $r_e(m) := s_e$ zum Zeitpunkt m.

- Ein *Durchgang* in einem System ist eine Funktion von den natürlichen Zahlen auf eine Folge von globalen Zuständen eines Systems.
- L_e :=Menge möglicher Umgebungszustände.
- L_i :=Menge möglicher lokaler Zustände von Agent i.
- $G := L_e x L_1 x ... x L_n$ Menge der globalen Zustände.
- Ein Paar (r, m) heißt 'Punkt' in einem System und bezeichnet Durchgang r zum Zeitpunkt m.
- $r(m) := (s_e, s_1, ..., s_n)$ zum Zeitpunkt m.
- $r_e(m) := s_e$ zum Zeitpunkt m.
- $r_i(m) := s_i$ zum Zeitpunkt m.

Definition

Ein System $\mathcal{R}$ über G ist eine nichtleere Menge von Durchgängen in G.

- Das Bit-Transmissions-Problem lässt sich nun folgendermaßen als System formalisieren:

- Das Bit-Transmissions-Problem lässt sich nun folgendermaßen als System formalisieren:
- $L_S = \{0, 1, (0, ack), (1, ack)\}$

- Das Bit-Transmissions-Problem lässt sich nun folgendermaßen als System formalisieren:
- $L_S = \{0, 1, (0, ack), (1, ack)\}$
- $L_R = \{\lambda, 0, 1\}$

- Das Bit-Transmissions-Problem lässt sich nun folgendermaßen als System formalisieren:
- $L_S = \{0, 1, (0, ack), (1, ack)\}$
- $L_R = \{\lambda, 0, 1\}$
- $L_e = A^*$
  $A = \{(sendbit, \Lambda), (\Lambda, sendack), (sendbit, sendack)\}$
  $= \{(a_1, ..., a_n) \mid n \geq 0; a_i \in A\}$

- Das Bit-Transmissions-Problem lässt sich nun folgendermaßen als System formalisieren:
- $L_S = \{0, 1, (0, ack), (1, ack)\}$
- $L_R = \{\lambda, 0, 1\}$
- $L_e = A^*$
  $A = \{(sendbit, \Lambda), (\Lambda, sendack), (sendbit, sendack)\}$
  $= \{(a_1, ..., a_n) \mid n \geq 0; a_i \in A\}$
- Das n-te Element bestimmt die Zustände von Sender und Empfänger in Runde n.

- Das Bit-Transmissions-Problem lässt sich nun folgendermaßen als System formalisieren:
- $L_S = \{0, 1, (0, ack), (1, ack)\}$
- $L_R = \{\lambda, 0, 1\}$
- $L_e = A^*$
  $A = \{(sendbit, \Lambda), (\Lambda, sendack), (sendbit, sendack)\}$
  $= \{(a_1, ..., a_n) \mid n \geq 0; a_i \in A\}$
- Das n-te Element bestimmt die Zustände von Sender und Empfänger in Runde n.
- $r(0) = (\varepsilon, k, \lambda), k \in \{0, 1\}$

- Das Bit-Transmissions-Problem lässt sich nun folgendermaßen als System formalisieren:
- $L_S = \{0, 1, (0, ack), (1, ack)\}$
- $L_R = \{\lambda, 0, 1\}$
- $L_e = A^*$
  $A = \{(sendbit, \Lambda), (\Lambda, sendack), (sendbit, sendack)\}$
  $= \{(a_1, ..., a_n) \mid n \geq 0; a_i \in A\}$
- Das n-te Element bestimmt die Zustände von Sender und Empfänger in Runde n.
- $r(0) = (\varepsilon, k, \lambda), k \in \{0, 1\}$
- $r(m) = (s_e, s_S, s_R)$

- Das Bit-Transmissions-Problem lässt sich nun folgendermaßen als System formalisieren:
- $L_S = \{0, 1, (0, ack), (1, ack)\}$
- $L_R = \{\lambda, 0, 1\}$
- $L_e = A^*$
  $A = \{(sendbit, \Lambda), (\Lambda, sendack), (sendbit, sendack)\}$
  $= \{(a_1, ..., a_n) \mid n \geq 0; a_i \in A\}$
- Das n-te Element bestimmt die Zustände von Sender und Empfänger in Runde n.
- $r(0) = (\varepsilon, k, \lambda), k \in \{0, 1\}$
- $r(m) = (s_e, s_S, s_R)$
- $r(m+1) = (s'_e, s'_S, s'_R)$

- Zusätzlich gelten folgende Bedingungen:

- Zusätzlich gelten folgende Bedingungen:
- wenn $s_R = \lambda$ dann $s'_S = s_S$ und $s'_e = s_e \cdot (sendbit, \Lambda)$ und $(s'_R = S_S$ oder $s'_R = \lambda)$

- Zusätzlich gelten folgende Bedingungen:
- wenn $s_R = \lambda$ dann $s'_S = s_S$ und $s'_e = s_e \cdot (sendbit, \Lambda)$ und $(s'_R = S_S$ oder $s'_R = \lambda)$
- wenn $s_S = s_R = k$ dann $s'_R = k, s'_S = s_S$ oder $s'_S = (k, ack), s'_e = s_e \cdot (sendbit, sendack)$ $(k \in \{0, 1\})$

- Zusätzlich gelten folgende Bedingungen:
- wenn $s_R = \lambda$ dann $s'_S = s_S$ und $s'_e = s_e \cdot (sendbit, \Lambda)$ und $(s'_R = S_S$ oder $s'_R = \lambda)$
- wenn $s_S = s_R = k$ dann $s'_R = k, s'_S = s_S$ oder $s'_S = (k, ack), s'_e = s_e \cdot (sendbit, sendack)$ $(k \in \{0, 1\})$
- wenn $s_S = (k, ack)$ dann $s'_e = s_e \cdot (\Lambda, sendack), s'_S = s_S$ und $s'_R = s_R$

- Das Wissen eines Agenten wird von seinem lokalen Zustand bestimmt.

- Das Wissen eines Agenten wird von seinem lokalen Zustand bestimmt.
- R weiß φ nicht: Soweit es R betrifft könnte das System in einem Zustand sein, in dem $\neg\varphi$ gilt.

- Das Wissen eines Agenten wird von seinem lokalen Zustand bestimmt.
- R weiß φ nicht: Soweit es R betrifft könnte das System in einem Zustand sein, in dem $\neg\varphi$ gilt.
- Sei Φ eine Menge von atomaren Aussagen.

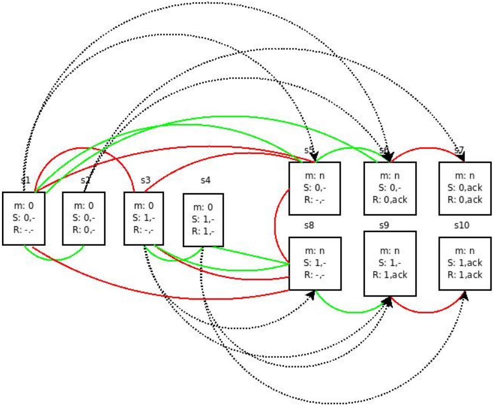
- Das Wissen eines Agenten wird von seinem lokalen Zustand bestimmt.
- R weiß φ nicht: Soweit es R betrifft könnte das System in einem Zustand sein, in dem $\neg\varphi$ gilt.
- Sei Φ eine Menge von atomaren Aussagen.
- Ein *interpretiertes System* $\mathcal{I}$ ist ein Paar $(\mathcal{R}, \pi)$, wo $\mathcal{R}$ ein System über $\mathcal{G}$ ist und π eine Interpretation für die Aussagen in Φ über $\mathcal{G}$.

- Um Wissen zu repräsentieren, assoziieren wir ein interpretiertes System $\mathcal{I} = (\mathcal{R}, \pi)$ mit einer Kripke-Struktur: $\mathcal{M}_{\mathcal{I}} = (\mathcal{S}, \pi, K_1, \ldots, K_n)$

- Um Wissen zu repräsentieren, assoziieren wir ein interpretiertes System $\mathcal{I} = (\mathcal{R}, \pi)$ mit einer Kripke-Struktur: $\mathcal{M}_{\mathcal{I}} = (\mathcal{S}, \pi, K_1, ..., K_n)$
- $\mathcal{S}$ besteht aus dem Punkten (r,m) in $\mathcal{I}$.

- Um Wissen zu repräsentieren, assoziieren wir ein interpretiertes System $\mathcal{I} = (\mathcal{R}, \pi)$ mit einer Kripke-Struktur: $\mathcal{M}_{\mathcal{I}} = (\mathcal{S}, \pi, K_1, ..., K_n)$
- $\mathcal{S}$ besteht aus dem Punkten (r,m) in $\mathcal{I}$.
- $K_1, ..., K_n$ sind binäre Relationen auf $\mathcal{S}$.

- Um Wissen zu repräsentieren, assoziieren wir ein interpretiertes System $\mathcal{I} = (\mathcal{R}, \pi)$ mit einer Kripke-Struktur: $\mathcal{M}_{\mathcal{I}} = (\mathcal{S}, \pi, K_1, ..., K_n)$
- $\mathcal{S}$ besteht aus dem Punkten (r,m) in $\mathcal{I}$.
- $K_1, ..., K_n$ sind binäre Relationen auf $\mathcal{S}$.
- Es gibt keine Relation K_e.


s1
m: 0
S: 0,-
R: -,-
s2
m: 0
S: 0,-
R: 0,-
s3
m: 0
S: 1,-
R: -,-
s4
m: 0
S: 1,-
R: 1,-
s5
m: n
S: 0,-
R: -,-
s6
m: n
S: 0,-
R: 0,ack
s7
m: n
S: 0,ack
R: 0,ack
s8
m: n
S: 1,-
R: -,-
s9
m: n
S: 1,-
R: 1,ack
s10
m: n
S: 1,ack
R: 1,ack

- Sind s=(s_e,s_1,...,s_n) und s'=(s'_e,s'_1,...,s'_n) globale Zustände in $\mathcal{R}$, so heißen s und s' *ununterscheidbar* für Agent i gdw:
  $s, s' \in K_i \wedge s_i = s'_i$
  Notation: $s \sim_i s'$

- Sind s=$(s_e,s_1,\dots,s_n)$ und s'=$(s'_e,s'_1,\dots,s'_n)$ globale Zustände in $\mathcal{R}$, so heißen s und s' *ununterscheidbar* für Agent i gdw:
  $s, s' \in K_i \wedge s_i = s'_i$
  Notation: $s \sim_i s'$
- Sind $r_i(m) = r'_i(m')$ so sind auch $(r, m) \sim_i (r', m')$

- Sind s=$(s_e,s_1,\ldots,s_n)$ und s'=$(s'_e,s'_1,\ldots,s'_n)$ globale Zustände in $\mathcal{R}$, so heißen s und s' *ununterscheidbar* für Agent i gdw:
  $s, s' \in K_i \wedge s_i = s'_i$
  Notation: $s \sim_i s'$
- Sind $r_i(m) = r'_i(m')$ so sind auch $(r, m) \sim_i (r', m')$
- Somit wird das Wissen eines Agenten i durch seinen lokalen Zustand determiniert.

- Sind s=(s_e,s_1,...,s_n) und s'=(s'_e,s'_1,...,s'_n) globale Zustände in $\mathcal{R}$, so heißen s und s' *ununterscheidbar* für Agent i gdw:
  $s, s' \in K_i \wedge s_i = s'_i$
  Notation: $s \sim_i s'$
- Sind $r_i(m) = r'_i(m')$ so sind auch $(r, m) \sim_i (r', m')$
- Somit wird das Wissen eines Agenten i durch seinen lokalen Zustand determiniert.
- Wir können nun bestimmen, was es für eine Formel $\varphi \in \mathcal{L}_n^{\mathcal{CD}}$ im Punkt (r, m) bedeutet, wahr zu sein.

- $(\mathcal{I}, r, m) \models \varphi (\varphi \in \Phi) \iff \pi(r, m)(\varphi) = \textbf{true}$

- $(\mathcal{I}, r, m) \models \varphi (\varphi \in \Phi) \iff \pi(r, m)(\varphi) = \textbf{true}$
- $(\mathcal{I}, r, m) \models K_i\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m') so dass $(r, m) \sim_i (r', m')$

- $(\mathcal{I}, r, m) \models \varphi(\varphi \in \Phi) \iff \pi(r, m)(\varphi) = \textbf{true}$
- $(\mathcal{I}, r, m) \models K_i\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m') so dass $(r, m) \sim_i (r', m')$
- $\mathcal{I} \models \varphi \iff (\mathcal{I}, r, m) \models \varphi$ für alle (r, m) in $\mathcal{I}$

- $(\mathcal{I}, r, m) \models \varphi (\varphi \in \Phi) \iff \pi(r, m)(\varphi) = \textbf{true}$
- $(\mathcal{I}, r, m) \models K_i\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m') so dass $(r, m) \sim_i (r', m')$
- $\mathcal{I} \models \varphi \iff (\mathcal{I}, r, m) \models \varphi$ für alle (r, m) in $\mathcal{I}$
- $(\mathcal{I}, r, m) \models C_G\varphi \iff (\mathcal{I}, r, m) \models E_G^k\varphi$.

- $(\mathcal{I}, r, m) \models \varphi (\varphi \in \Phi) \iff \pi(r, m)(\varphi) = \textbf{true}$
- $(\mathcal{I}, r, m) \models K_i\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m') so dass $(r, m) \sim_i (r', m')$
- $\mathcal{I} \models \varphi \iff (\mathcal{I}, r, m) \models \varphi$ für alle (r, m) in $\mathcal{I}$
- $(\mathcal{I}, r, m) \models C_G\varphi \iff (\mathcal{I}, r, m) \models E_G^k\varphi$.
- $(\mathcal{I}, r), m \models E_G^k\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m'), die von (r, m) aus in k Schritten G-erreichbar sind (G ist die Menge aller Agenten).

- $(\mathcal{I}, r, m) \models \varphi (\varphi \in \Phi) \iff \pi(r, m)(\varphi) = \textbf{true}$
- $(\mathcal{I}, r, m) \models K_i\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m') so dass $(r, m) \sim_i (r', m')$
- $\mathcal{I} \models \varphi \iff (\mathcal{I}, r, m) \models \varphi$ für alle (r, m) in $\mathcal{I}$
- $(\mathcal{I}, r, m) \models C_G\varphi \iff (\mathcal{I}, r, m) \models E_G^k\varphi$.
- $(\mathcal{I}, r), m \models E_G^k\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m'), die von (r, m) aus in k Schritten G-erreichbar sind (G ist die Menge aller Agenten).
- $(\mathcal{I}, r, m) \models D_G\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m') mit $((r, m)(r', m')) \in \cap_{i \in G} K_i$

- R^{bt} besteht aus der Menge von Durchgängen, aus dem vorigen Beispiel und π^{bt} gestaltet sich folgerndermaßen:

- R^{bt} besteht aus der Menge von Durchgängen, aus dem vorigen Beispiel und π^{bt} gestaltet sich folgerndermaßen:
- $\Phi = \{bit = 0, bit = 1, recbit, recack, sendbit, sendack\}$

- R^{bt} besteht aus der Menge von Durchgängen, aus dem vorigen Beispiel und π^{bt} gestaltet sich folgerndermaßen:
- $\Phi = \{bit = 0, bit = 1, recbit, recack, sendbit, sendack\}$
- $(\mathcal{I}^{bt}, r, m) \models bit = k \iff$
  $r_S(m) = k \vee r_S(m) = (k, ack)$ für $k = 0, 1$

- R^{bt} besteht aus der Menge von Durchgängen, aus dem vorigen Beispiel und π^{bt} gestaltet sich folgerndermaßen:
- $\Phi = \{bit = 0, bit = 1, recbit, recack, sendbit, sendack\}$
- $(\mathcal{I}^{bt}, r, m) \models bit = k \iff$
  $r_S(m) = k \vee r_S(m) = (k, ack)$ für $k = 0, 1$
- $(\mathcal{I}^{bt}, r, m) \models recbit \iff r_R(m) \neq \lambda$

- R^{bt} besteht aus der Menge von Durchgängen, aus dem vorigen Beispiel und π^{bt} gestaltet sich folgerndermaßen:
- $\Phi = \{bit = 0, bit = 1, recbit, recack, sendbit, sendack\}$
- $(\mathcal{I}^{bt}, r, m) \models bit = k \iff$
  $r_S(m) = k \vee r_S(m) = (k, ack)$ für $k = 0, 1$
- $(\mathcal{I}^{bt}, r, m) \models recbit \iff r_R(m) \neq \lambda$
- $(\mathcal{I}^{bt}, r, m) \models recack \iff r_S(m) = (0, ack) \vee r_S(m) = (1, ack)$

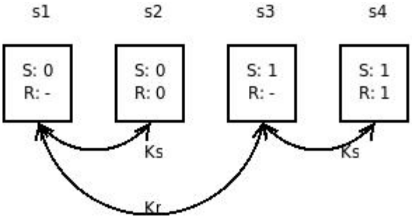
- R^{bt} besteht aus der Menge von Durchgängen, aus dem vorigen Beispiel und π^{bt} gestaltet sich folgerndermaßen:
- $\Phi = \{bit = 0, bit = 1, recbit, recack, sendbit, sendack\}$
- $(\mathcal{I}^{bt}, r, m) \models bit = k \iff$
  $r_S(m) = k \vee r_S(m) = (k, ack)$ für $k = 0, 1$
- $(\mathcal{I}^{bt}, r, m) \models recbit \iff r_R(m) \neq \lambda$
- $(\mathcal{I}^{bt}, r, m) \models recack \iff r_S(m) = (0, ack) \vee r_S(m) = (1, ack)$
- $(\mathcal{I}^{bt}, r, m) \models sendbit$ wenn das letzte Tupel in $r_e(m)$ entweder $(sendbit, \Lambda)$ oder $(\Lambda, sendack)$ ist.

- R^{bt} besteht aus der Menge von Durchgängen, aus dem vorigen Beispiel und π^{bt} gestaltet sich folgerndermaßen:
- $\Phi = \{bit = 0, bit = 1, recbit, recack, sendbit, sendack\}$
- $(\mathcal{I}^{bt}, r, m) \models bit = k \iff$ $r_S(m) = k \lor r_S(m) = (k, ack)$ für $k = 0, 1$
- $(\mathcal{I}^{bt}, r, m) \models recbit \iff r_R(m) \neq \lambda$
- $(\mathcal{I}^{bt}, r, m) \models recack \iff r_S(m) = (0, ack) \lor r_S(m) = (1, ack)$
- $(\mathcal{I}^{bt}, r, m) \models sendbit$ wenn das letzte Tupel in $r_e(m)$ entweder $(sendbit, \Lambda)$ oder $(\Lambda, sendack)$ ist.
- $(\mathcal{I}^{bt}, r, m) \models sendack$ wenn das letzte Tupel in $r_e(m)$ entweder $(\Lambda, sendack)$ oder $(sendbit, sendack)$ ist.

- Intuitiv gilt: Nachdem R das Bit von S empfangen hat, weiß R den Wert des Bits. Und in der Tat:

- Intuitiv gilt: Nachdem R das Bit von S empfangen hat, weiß R den Wert des Bits. Und in der Tat:
- $(\mathcal{I}^{bt}, r, m) \models K_R(bit = k), k \in \{0, 1\}$, da für jeden Punkt $(r', m') gilt:$

- Intuitiv gilt: Nachdem R das Bit von S empfangen hat, weiß R den Wert des Bits. Und in der Tat:
- $(\mathcal{I}^{bt}, r, m) \models K_R(bit = k), k \in \{0, 1\}$, da für jeden Punkt $(r', m')gilt:$
- wenn $r_R(m) = r'_R(m')$ dann muss S bit k gesendet haben:

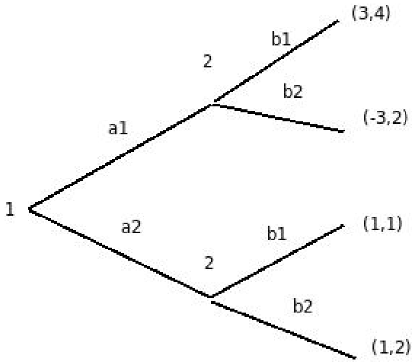
- Intuitiv gilt: Nachdem R das Bit von S empfangen hat, weiß R den Wert des Bits. Und in der Tat:
- $(\mathcal{I}^{bt}, r, m) \models K_R(bit = k), k \in \{0, 1\}$, da für jeden Punkt $(r', m') gilt:$
- wenn $r_R(m) = r'_R(m')$ dann muss S bit k gesendet haben:
- 

- Spiel 'G_1': Spieler 1 und Spieler 2 ziehen abwechselnd.

- Spiel 'G_1': Spieler 1 und Spieler 2 ziehen abwechselnd.
- Spieler 1 beginnt und hat die Wahl zwischen den Aktionen $a1$ und $a2$.

- Spiel 'G_1': Spieler 1 und Spieler 2 ziehen abwechselnd.
- Spieler 1 beginnt und hat die Wahl zwischen den Aktionen $a1$ und $a2$.
- Spieler 2 sieht die Aktion, die Spieler 1 gemacht hat und wählt dann zwischen $b1$ und $b2$.

- Spiel 'G_1': Spieler 1 und Spieler 2 ziehen abwechselnd.
- Spieler 1 beginnt und hat die Wahl zwischen den Aktionen $a1$ und $a2$.
- Spieler 2 sieht die Aktion, die Spieler 1 gemacht hat und wählt dann zwischen $b1$ und $b2$.
- Danach gibt es eine Auszahlung.


(3,4)
b1
2
b2
(-3,2)
a1
1
a2
b1
(1,1)
2
b2
(1,2)

- Ein solches Spiel heißt Spiel mit vollständiger Information, weil jeder Zug öffentlich geschieht.

- Ein solches Spiel heißt Spiel mit vollständiger Information, weil jeder Zug öffentlich geschieht.
- Genau genommen ist jeder Zug *Common knowledge* sobald er vollzogen wird:

- Ein solches Spiel heißt Spiel mit vollständiger Information, weil jeder Zug öffentlich geschieht.
- Genau genommen ist jeder Zug *Common knowledge* sobald er vollzogen wird:
- $\Phi = act_i(a), a \in \{a1, a2, b1, b2\}$ und $i \in \{1, 2\}$

- Ein solches Spiel heißt Spiel mit vollständiger Information, weil jeder Zug öffentlich geschieht.
- Genau genommen ist jeder Zug *Common knowledge* sobald er vollzogen wird:
- $\Phi = act_i(a), a \in \{a1, a2, b1, b2\}$ und $i \in \{1, 2\}$
- Sei $\mathcal{I}_1 = (\mathcal{R}_1, \pi)$ ein interpretiertes System, das mit dem Spiel G_1 korrespondiert, so dass:

- Ein solches Spiel heißt Spiel mit vollständiger Information, weil jeder Zug öffentlich geschieht.
- Genau genommen ist jeder Zug *Common knowledge* sobald er vollzogen wird:
- $\Phi = act_i(a), a \in \{a1, a2, b1, b2\}$ und $i \in \{1, 2\}$
- Sei $\mathcal{I}_1 = (\mathcal{R}_1, \pi)$ ein interpretiertes System, das mit dem Spiel G_1 korrespondiert, so dass:
- $\pi(r, m)(act_i(a)) = \textbf{true}$ gdw. i hat im globalen Zustand $r(m)$ Aktion a getätigt. Dann gilt:

- (a) $(\mathcal{I}_1, r, 1) \models act_1(a) \iff C(act_1(a))$

- (a) $(\mathcal{I}_1, r, 1) \models act_1(a) \iff C(act_1(a))$
- (b) $(\mathcal{I}_1, r, 2) \models act_2(a) \iff C(act_2(a))$

- (a) $(\mathcal{I}_1, r, 1) \models act_1(a) \iff C(act_1(a))$:

- (a) $(\mathcal{I}_1, r, 1) \models act_1(a) \iff C(act_1(a))$:
- $(\mathcal{I}, r, m) \models C_G\varphi \iff (\mathcal{I}, r, m) \models E_G^k\varphi$.

- (a) $(\mathcal{I}_1, r, 1) \models act_1(a) \iff C(act_1(a))$:
- $(\mathcal{I}, r, m) \models C_G\varphi \iff (\mathcal{I}, r, m) \models E_G^k\varphi$.
- $(\mathcal{I}, r), m \models E_G^k\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m'), die von (r, m) aus in k Schritten G-erreichbar sind (G ist die Menge aller Agenten).

- (a) $(\mathcal{I}_1, r, 1) \models act_1(a) \iff C(act_1(a))$:
- $(\mathcal{I}, r, m) \models C_G\varphi \iff (\mathcal{I}, r, m) \models E_G^k\varphi$.
- $(\mathcal{I}, r), m \models E_G^k\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m'), die von (r, m) aus in k Schritten G-erreichbar sind (G ist die Menge aller Agenten).
- $r(1) = (act_1(a1) \wedge K_1(act_1(a1) \wedge K_2(act_1(a_1))$
  $r'(1) = (act_1(a2) \wedge K_1(act_1(a2)) \wedge K_2(act_1(a2))$

- (a) $(\mathcal{I}_1, r, 1) \models act_1(a) \iff C(act_1(a))$:
- $(\mathcal{I}, r, m) \models C_G\varphi \iff (\mathcal{I}, r, m) \models E_G^k\varphi$.
- $(\mathcal{I}, r), m \models E_G^k\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m'), die von (r, m) aus in k Schritten G-erreichbar sind (G ist die Menge aller Agenten).
- $r(1) = (act_1(a1) \wedge K_1(act_1(a1) \wedge K_2(act_1(a_1))$
  $r'(1) = (act_1(a2) \wedge K_1(act_1(a2)) \wedge K_2(act_1(a2))$
- $K_1 = \{(r(1), r(1)), (r'(1), r'(1))\}$

- (a) $(\mathcal{I}_1, r, 1) \models act_1(a) \iff C(act_1(a))$:
- $(\mathcal{I}, r, m) \models C_G\varphi \iff (\mathcal{I}, r, m) \models E_G^k\varphi$.
- $(\mathcal{I}, r), m \models E_G^k\varphi \iff (\mathcal{I}, r', m') \models \varphi$ für alle (r', m'), die von (r, m) aus in k Schritten G-erreichbar sind (G ist die Menge aller Agenten).
- $r(1) = (act_1(a1) \wedge K_1(act_1(a1) \wedge K_2(act_1(a_1))$
  $r'(1) = (act_1(a2) \wedge K_1(act_1(a2)) \wedge K_2(act_1(a2))$
- $K_1 = \{(r(1), r(1)), (r'(1), r'(1))\}$
- $K_2 = \{(r(1), r(1)), (r'(1), r'(1))\}$

- Um temporale Ausdrücke zuzulassen, wird die Sprache um die folgenden Operatoren erweitert:

- Um temporale Ausdrücke zuzulassen, wird die Sprache um die folgenden Operatoren erweitert:
- $(\mathcal{I}, r, m) \models \Box\varphi \Longleftrightarrow (\mathcal{I}, r, m') \models \varphi$ für alle $m' \geq m$

- Um temporale Ausdrücke zuzulassen, wird die Sprache um die folgenden Operatoren erweitert:

- $(\mathcal{I}, r, m) \models \Box\varphi \Longleftrightarrow (\mathcal{I}, r, m') \models \varphi$ für alle $m' \geq m$
- $(\mathcal{I}, r, m) \models \Diamond\varphi \Longleftrightarrow (\mathcal{I}, r, m') \models \varphi$ für mindestens ein $m' \geq m$

- Um temporale Ausdrücke zuzulassen, wird die Sprache um die folgenden Operatoren erweitert:

- $(\mathcal{I}, r, m) \models \Box\varphi \iff (\mathcal{I}, r, m') \models \varphi$ für alle $m' \geq m$
- $(\mathcal{I}, r, m) \models \Diamond\varphi \iff (\mathcal{I}, r, m') \models \varphi$ für mindestens ein $m' \geq m$
- $(\mathcal{I}, r, m) \models \bigcirc\varphi \iff (\mathcal{I}, r, m+1) \models \varphi$

- Um temporale Ausdrücke zuzulassen, wird die Sprache um die folgenden Operatoren erweitert:

- $(\mathcal{I}, r, m) \models \Box\varphi \iff (\mathcal{I}, r, m') \models \varphi$ für alle $m' \geq m$
- $(\mathcal{I}, r, m) \models \Diamond\varphi \iff (\mathcal{I}, r, m') \models \varphi$ für mindestens ein $m' \geq m$
- $(\mathcal{I}, r, m) \models \bigcirc\varphi \iff (\mathcal{I}, r, m+1) \models \varphi$
- $(\mathcal{I}, r, m) \models \varphi U \psi \iff (\mathcal{I}, r, m') \models \psi$
  für mindestens ein $m' \geq m$ und $(\mathcal{I}, r, m'') \models \varphi$ für alle m'' mit $m \leq m'' < m'$

- Eine temporale Formel ist eine Formel, deren einzige Modaloperatoren temporale Operatoren sind.

- Eine temporale Formel ist eine Formel, deren einzige Modaloperatoren temporale Operatoren sind.
- Eine Wissensformel ist eine Formel, deren einzige Modaloperatoren $K_1, \ldots, K_n$, C_G und D_G sind.

- Eine temporale Formel ist eine Formel, deren einzige Modaloperatoren temporale Operatoren sind.
- Eine Wissensformel ist eine Formel, deren einzige Modaloperatoren $K_1, ..., K_n$, C_G und D_G sind.
- Wir haben gesehen, dass zwei Punkte (r, m),(r', m') in einem interpretierten System $\mathcal{I}$ dann gleich sind, wenn alle Formeln $\varphi \in \mathcal{L}_n^{\mathcal{CD}}$ dieselben Wahrheitswerte unter π zurückgeben.

- Eine temporale Formel ist eine Formel, deren einzige Modaloperatoren temporale Operatoren sind.
- Eine Wissensformel ist eine Formel, deren einzige Modaloperatoren $K_1, \ldots, K_n$, C_G und D_G sind.
- Wir haben gesehen, dass zwei Punkte (r, m),(r', m') in einem interpretierten System $\mathcal{I}$ dann gleich sind, wenn alle Formeln $\varphi \in \mathcal{L}_n^{\mathcal{CD}}$ dieselben Wahrheitswerte unter π zurückgeben.
- Dies gilt jedoch nur für Wissensformeln.

- Die Wahrheitswerte temporaler Formeln hängen nicht vom System, sondern nur vom jeweiligen Durchgang ab.

- Die Wahrheitswerte temporaler Formeln hängen nicht vom System, sondern nur vom jeweiligen Durchgang ab.
- Ist φ eine temporale Formel, so ist ihr Wahrheitswert am Punkt (r, m) in einem interpretierten System $(\mathcal{R}, \pi)$ nicht von $\mathcal{R}$ abhängig, sondern nur von π.

- Die Wahrheitswerte temporaler Formeln hängen nicht vom System, sondern nur vom jeweiligen Durchgang ab.
- Ist φ eine temporale Formel, so ist ihr Wahrheitswert am Punkt (r, m) in einem interpretierten System $(\mathcal{R}, \pi)$ nicht von $\mathcal{R}$ abhängig, sondern nur von π.
- r erfüllt φ wenn gilt: $(\pi, r, 0) \models \varphi$

- Die Wahrheitswerte temporaler Formeln hängen nicht vom System, sondern nur vom jeweiligen Durchgang ab.
- Ist φ eine temporale Formel, so ist ihr Wahrheitswert am Punkt (r, m) in einem interpretierten System $(\mathcal{R}, \pi)$ nicht von $\mathcal{R}$ abhängig, sondern nur von π.
- r erfüllt φ wenn gilt: $(\pi, r, 0) \models \varphi$
- Temporale Formeln werden verwendet, um über Ereignisse zu sprechen, die während eines einzelnen Durchgangs geschehen.

- Die Wahrheitswerte temporaler Formeln hängen nicht vom System, sondern nur vom jeweiligen Durchgang ab.
- Ist φ eine temporale Formel, so ist ihr Wahrheitswert am Punkt (r, m) in einem interpretierten System $(\mathcal{R}, \pi)$ nicht von $\mathcal{R}$ abhängig, sondern nur von π.
- r erfüllt φ wenn gilt: $(\pi, r, 0) \models \varphi$
- Temporale Formeln werden verwendet, um über Ereignisse zu sprechen, die während eines einzelnen Durchgangs geschehen.
- Beispiel: $\Box(\mathit{recbit} \rightarrow \Diamond \mathit{recack})$

- Dadurch, dass wir temporale und Wissensformeln kombinieren, können wir Aussagen über die Entwicklung des Wissens in einem System machen.

- Dadurch, dass wir temporale und Wissensformeln kombinieren, können wir Aussagen über die Entwicklung des Wissens in einem System machen.
- Die Aussage *Irgendwann weiß der Empfänger das erste Bit des Senders* ließe sich wie folgt repräsentieren:

$$\Diamond(K_R(bit = 0) \vee K_R(bit = 1))$$

- Eine knowledge base (KB) ist ein System, dem Fakten über eine externe Welt erzählt werden.

- Eine knowledge base (KB) ist ein System, dem Fakten über eine externe Welt erzählt werden.
- Außerdem kann man der KB Fragen stellen, woraufhin die KB errechnet, ob die Anfrage aus dem Wissen folgt, das ihr erzählt wurde.

- Eine knowledge base (KB) ist ein System, dem Fakten über eine externe Welt erzählt werden.
- Außerdem kann man der KB Fragen stellen, woraufhin die KB errechnet, ob die Anfrage aus dem Wissen folgt, das ihr erzählt wurde.
- Mit Hilfe des obigen Frameworks lässt sich modellieren, wie eine KB ihr Wissen erlangt und man kann dieses Wissen in Beziehng zu einer externen Welt setzen.

- Eine knowledge base (KB) ist ein System, dem Fakten über eine externe Welt erzählt werden.
- Außerdem kann man der KB Fragen stellen, woraufhin die KB errechnet, ob die Anfrage aus dem Wissen folgt, das ihr erzählt wurde.
- Mit Hilfe des obigen Frameworks lässt sich modellieren, wie eine KB ihr Wissen erlangt und man kann dieses Wissen in Beziehng zu einer externen Welt setzen.
- Eine KB sei ein Agent. Obendrein gibt es einen *Erzähler*, der der KB ausgewählte Fakten über die externe Welt erzählt.

- Zusätzlich gelten folgende Annahmen:

- Zusätzlich gelten folgende Annahmen:
  1. Die externe Welt lässt sich als eine Menge von Propositionen Φ darstellen.

- Zusätzlich gelten folgende Annahmen:
  1. Die externe Welt lässt sich als eine Menge von Propositionen Φ darstellen.
  2. Die externe Welt ist stabil, so dass sich die Wahrheitswerte der $\varphi \in \Phi$ nicht mit der Zeit ändern.

- Zusätzlich gelten folgende Annahmen:
  1. Die externe Welt lässt sich als eine Menge von Propositionen Φ darstellen.
  2. Die externe Welt ist stabil, so dass sich die Wahrheitswerte der $\varphi \in \Phi$ nicht mit der Zeit ändern.
  3. Der Erzähler weiß alles über die externe Welt.

- Zusätzlich gelten folgende Annahmen:
  1. Die externe Welt lässt sich als eine Menge von Propositionen Φ darstellen.
  2. Die externe Welt ist stabil, so dass sich die Wahrheitswerte der $\varphi \in \Phi$ nicht mit der Zeit ändern.
  3. Der Erzähler weiß alles über die externe Welt.
  4. Der KB werden nur Fakten über die externe Welt in Form von propositionalen Formeln erzählt und keine Aussagen über ihr eigenes Wissen.

- Zusätzlich gelten folgende Annahmen:
  1. Die externe Welt lässt sich als eine Menge von Propositionen Φ darstellen.
  2. Die externe Welt ist stabil, so dass sich die Wahrheitswerte der $\varphi \in \Phi$ nicht mit der Zeit ändern.
  3. Der Erzähler weiß alles über die externe Welt.
  4. Der KB werden nur Fakten über die externe Welt in Form von propositionalen Formeln erzählt und keine Aussagen über ihr eigenes Wissen.
  5. Alles was der KB erzählt wird ist wahr.

- Zusätzlich gelten folgende Annahmen:
  1. Die externe Welt lässt sich als eine Menge von Propositionen Φ darstellen.
  2. Die externe Welt ist stabil, so dass sich die Wahrheitswerte der $\varphi \in \Phi$ nicht mit der Zeit ändern.
  3. Der Erzähler weiß alles über die externe Welt.
  4. Der KB werden nur Fakten über die externe Welt in Form von propositionalen Formeln erzählt und keine Aussagen über ihr eigenes Wissen.
  5. Alles was der KB erzählt wird ist wahr.
  6. Es gibt kein *a priorisches* Wissen über die externe Welt oder das, was der KB erzählt wird.

- Auf das Systemmodell übertragen bedeutet das:
  1. Der Umgebungszustand s_e lässt sich als Wahrheitsfunktion $\alpha(\Phi)$ darstellen.

- Auf das Systemmodell übertragen bedeutet das:
  1. Der Umgebungszustand s_e lässt sich als Wahrheitsfunktion $\alpha(\Phi)$ darstellen.
  2. Bei jedem Durchgang r ist der Umgebumgszustand $r_e(m)$ unabhängig von m.

- Auf das Systemmodell übertragen bedeutet das:
  1. Der Umgebungszustand s_e lässt sich als Wahrheitsfunktion $\alpha(\Phi)$ darstellen.
  2. Bei jedem Durchgang r ist der Umgebumgszustand $r_e(m)$ unabhängig von m.
  3. Der lokale Zustand des Erzählers beinhaltet $\alpha(\Phi)$.

- Auf das Systemmodell übertragen bedeutet das:
  1. Der Umgebungszustand s_e lässt sich als Wahrheitsfunktion $\alpha(\Phi)$ darstellen.
  2. Bei jedem Durchgang r ist der Umgebumgszustand $r_e(m)$ unabhängig von m.
  3. Der lokale Zustand des Erzählers beinhaltet $\alpha(\Phi)$.
  4. Der lokale Zustand der KB hat die Form $<\varphi_1, ..., \varphi_k>$, $k \geq 0$.

- Auf das Systemmodell übertragen bedeutet das:
  1. Der Umgebungszustand s_e lässt sich als Wahrheitsfunktion $\alpha(\Phi)$ darstellen.
  2. Bei jedem Durchgang r ist der Umgebumgszustand $r_e(m)$ unabhängig von m.
  3. Der lokale Zustand des Erzählers beinhaltet $\alpha(\Phi)$.
  4. Der lokale Zustand der KB hat die Form $<\varphi_1, ..., \varphi_k>$, $k \geq 0$.
  5. Im globalen Zustand $s_G = (\alpha, < \varphi_1, ..., \varphi_k >, \cdot)$ ist jedes φ wahr unter α.

- Auf das Systemmodell übertragen bedeutet das:
  1. Der Umgebungszustand s_e lässt sich als Wahrheitsfunktion $\alpha(\Phi)$ darstellen.
  2. Bei jedem Durchgang r ist der Umgebumgszustand $r_e(m)$ unabhängig von m.
  3. Der lokale Zustand des Erzählers beinhaltet $\alpha(\Phi)$.
  4. Der lokale Zustand der KB hat die Form $<\varphi_1, ..., \varphi_k>$, $k \geq 0$.
  5. Im globalen Zustand $s_G = (\alpha, < \varphi_1, ..., \varphi_k >, \cdot)$ ist jedes φ wahr unter α.
  6. r(0) hat die Form $(\alpha, <>, \cdot)$.

- $\mathcal{I}^{kb} = (\mathcal{R}^{kb}, \pi^{kb})$, wobei $\mathcal{R}^{kb}$ aus allen Durchgängen besteht, bei denen für eine Folge $\varphi_1, ..., \varphi_k$ von propositionalen Formeln und eine Wahrheitsfunktion α gilt:

- $\mathcal{I}^{kb} = (\mathcal{R}^{kb}, \pi^{kb})$, wobei $\mathcal{R}^{kb}$ aus allen Durchgängen besteht, bei denen für eine Folge $\varphi_1, ..., \varphi_k$ von propositionalen Formeln und eine Wahrheitsfunktion α gilt:
- $r(0) = (\alpha, <>, \cdot)$.

- $\mathcal{I}^{kb} = (\mathcal{R}^{kb}, \pi^{kb})$, wobei $\mathcal{R}^{kb}$ aus allen Durchgängen besteht, bei denen für eine Folge $\varphi_1, ..., \varphi_k$ von propositionalen Formeln und eine Wahrheitsfunktion α gilt:
- $r(0) = (\alpha, <>, \cdot)$.
- Wenn $r(m) = (\alpha, < \varphi_1, ..., \varphi_k >, \cdot)$.
  Dann gilt:

- $\mathcal{I}^{kb} = (\mathcal{R}^{kb}, \pi^{kb})$, wobei $\mathcal{R}^{kb}$ aus allen Durchgängen besteht, bei denen für eine Folge $\varphi_1, ..., \varphi_k$ von propositionalen Formeln und eine Wahrheitsfunktion α gilt:
- $r(0) = (\alpha, <>, \cdot)$.
- Wenn $r(m) = (\alpha, < \varphi_1, ..., \varphi_k >, \cdot)$.
  Dann gilt:
  1. entweder $r(m+1)) = r(m)$ oder
     $r(m+1) = (\alpha, < \varphi_1, ..., \varphi_k, \varphi_{k+1} >, \cdot)$.

- $\mathcal{I}^{kb} = (\mathcal{R}^{kb}, \pi^{kb})$, wobei $\mathcal{R}^{kb}$ aus allen Durchgängen besteht, bei denen für eine Folge $\varphi_1, ..., \varphi_k$ von propositionalen Formeln und eine Wahrheitsfunktion α gilt:
- $r(0) = (\alpha, <>, \cdot)$.
- Wenn $r(m) = (\alpha, < \varphi_1, ..., \varphi_k >, \cdot)$.
  Dann gilt:
  1. entweder $r(m+1)) = r(m)$ oder
     $r(m+1) = (\alpha, < \varphi_1, ..., \varphi_k, \varphi_{k+1} >, \cdot)$.
  2. $\varphi_1 \wedge ... \wedge \varphi_k$ ist wahr unter α.

- $\mathcal{I}^{kb} = (\mathcal{R}^{kb}, \pi^{kb})$, wobei $\mathcal{R}^{kb}$ aus allen Durchgängen besteht, bei denen für eine Folge $\varphi_1, ..., \varphi_k$ von propositionalen Formeln und eine Wahrheitsfunktion α gilt:
- $r(0) = (\alpha, <>, \cdot)$.
- Wenn $r(m) = (\alpha, < \varphi_1, ..., \varphi_k >, \cdot)$.
  Dann gilt:
  1. entweder $r(m+1)) = r(m)$ oder
     $r(m+1) = (\alpha, < \varphi_1, ..., \varphi_k, \varphi_{k+1} >, \cdot)$.
  2. $\varphi_1 \wedge ... \wedge \varphi_k$ ist wahr unter α.
  3. $\pi^{kb}(r, m) = \alpha$, also π^{kb} ist so definiert, dass die Wahrheitsfunktion an Punkt (r, m) durch s_e gegeben ist.

- Unsere KB kann nun folgendermaßen auf Fragen antworten: Wird ihr im Punkt (r, m) Anfrage ψ gestellt, antwortet sie:

- Unsere KB kann nun folgendermaßen auf Fragen antworten: Wird ihr im Punkt (r, m) Anfrage ψ gestellt, antwortet sie:
- 'Ja' gdw. $(\mathcal{I}^{kb}, r, m) \models K_{kb}\psi$

- Unsere KB kann nun folgendermaßen auf Fragen antworten: Wird ihr im Punkt (r, m) Anfrage ψ gestellt, antwortet sie:
- 'Ja' gdw. $(\mathcal{I}^{kb}, r, m) \models K_{kb}\psi$
- 'Nein' gdw. $(\mathcal{I}^{kb}, r, m) \models K_{kb}\neg\psi$

- Unsere KB kann nun folgendermaßen auf Fragen antworten: Wird ihr im Punkt (r, m) Anfrage ψ gestellt, antwortet sie:
- 'Ja' gdw. $(\mathcal{I}^{kb}, r, m) \models K_{kb}\psi$
- 'Nein' gdw. $(\mathcal{I}^{kb}, r, m) \models K_{kb}\neg\psi$
- 'Ich weiß es nicht' sonst.

Satz

Sei $r_{KB}(m) =< \varphi_1, ..., \varphi_k >$ und $\kappa = \varphi_1 \wedge ... \wedge \varphi_k$ und ψ eine Anfrage. Die folgenden Sätze sind äquivalent:

1. $(\mathcal{I}^{kb}, r, m) \models K_{KB}\psi$
2. $\kappa \rightarrow \psi$ ist eine AL Tautologie
3. $\mathcal{M}_n^{rst} \models K_{KB}\kappa \rightarrow K_{KB}\psi$